**CE-158 Commands**SETCOM 2400,8,N,1 Set baud and parameters  
SETDEV xx Redirect IO to RS-232, CO=Cassette Out, CI=Cassette In  
PRINT COM$ Show com parameters  
PRINT DEV$ Show active redirection

**Saving and Loading BASIC as ASCII over RS-232**  
Line numbers must be padded with preceding spaces, 20h, so they take up exactly five characters. You need a two second delay between lines to give the PC-1500 time to tokenize the line. The PC-1500 uses CR, 0Dh, as a line ending. Sending CRLF will not work. When saving as ASCII there is a bug that ads an extra space after a REM statement.

**To Save BASIC as ASCII over RS-232**  
SETCOM 2400,8,N,1 Set baud and parameters (if not already done)  
OUTSTAT 0 Set handshaking lines on (if not already done)  
SETDEV CO Redirect cassette out to RS-232  
CSAVEa Save BASIC  
  
**To Load BASIC as ASCII over RS-232**  
SETCOM 2400,8,N,1 Set baud and parameters (if not already done)  
OUTSTAT 0 Set handshaking lines on (if not already done)  
SETDEV CI Redirect cassette input to RS-232  
CLOADa Load BASIC

**To Save Binary over RS-232**  
SETCOM 2400,8,N,1 Set baud and parameters (if not already done)  
OUTSTAT 0 Set handshaking lines on (if not already done)  
SETDEV CO Redirect cassette out to RS-232  
CSAVEM “name”;&0000,&3FFF Save BASIC

**To use LPRINT over RS232**  
SETCOM 2400,8,N,1 Set baud and parameters (if not already done)  
OUTSTAT 0 Set handshaking lines on (if not already done)  
SETDEV PO Redirect LPRINT output to RS232  
LPRINT “TEST” Sends only ASCII text ‘TEST’

**To use PRINT# over RS232**  
SETCOM 2400,8,N,1 Set baud and parameters (if not already done)  
OUTSTAT 0 Set handshaking lines on (if not already done)  
SETDEV C0 Redirect cassette out to RS-232  
PRINT#”FILENAME”;A$ can also do PRINT# F$;A$

**Programming notes**  
The PV and PU opcode values are swapped in the manual.

**Interfacing notes**

Pin 44 on 60pin expansion connector is FGND not VBAT as shown in the manual.

The mixed use of PU, PV, PU0, PU1 in the manual is a bit confusing. They are used as chip select signals.  
PU == PU0  
PV == PU1

**CE-158 Notes**

Dumping the CE-158 ROM  
  
The CE-158 has a 16K ROM in an 8K address space from 8000h-A000h. The CE-158 is selected when the PV signal is LOW, and the PU signal is used to select which half of the ROM to access.

1. Change start of BASIC to reserve 8K from 0000h (how)
2. Using ML Program
   1. Set PV to LOW
   2. Set PU to LOW
   3. Block copy from 8000h-A000h to 0000h-2000h
3. Transmit out over RS-232 as space delimited value
4. Using ML Program
   1. Set PV to LOW
   2. Set PU to HI
   3. Block copy from 8000h-A000h to 0000h-2000
5. Transmit out over RS-232 as space delimited value

PV and PU Signals

Opcodes: SPU=E1, RPU=E3, SPV=A8, RPV=B8  
  
PU PV Toggle tests  
SPU causes low to high transition, SPV causes low to high transition  
RPU causes high to low transition, RPV causes high to low transition

Registers

C - Carry flag set if Bit 7 High  
H - Half carry flag set if Bit 3 high  
Z - Zero flag set if result is zero  
V – Overflow flag set when result exceeds 8 bits

Note from Christophe Gottheimer on V register

There is a fourth register called MN or (V). But for some unknown reason, this CPU seems to work as follows: The N (VL) register is always 0 and this is why you can see in some code (like mine) the instruction &34 = LDA N (VL) = CLA that set the accumulator to 0 in one instruction better and faster than &B5 &00 = LDA &00.

The register seems to be loaded by the last upper byte when a 16-bits instruction is realized: POP R, PUSH R, CALL nn, etc. I never really find an interesting usage of this register. I very often use optimization with N (VL) . Also, POP MN (V) is a way to recall the last 16-bits word stored into the stack without losing any registers. In the SHARP's ROM, there are no examples of the usage for these 2 registers.

CE-158 IO D00H~D00F 16 bytes  
CE-158 UART D200~D203 4 Bytes  
CE-158 Interrupt Port address DE00~DFFF

INSTAT the same?

OUTSTAT Bit Function Direction  
7-5 N/A N/A  
4 DSR IN (In on CE-158)  
3 (D)CD IN (In on CE-158)  
2 CTS IN (In on CE-158)  
1 RTS OUT  
0 DTR OUT

CE-158 UART Registers D200 – D204  
D200 Data Write UART\_DATA\_W  
D201 Register Write UART\_REG\_W  
D202 Data Read UART\_DATA\_R  
D203 Register Read UART\_REG\_R

CE-158 LH5811 Registers D000 – D00F

SETCOM Address: 7858  
Command Example: SETCOM 300,8,N,1   
BAUD: 50, 100, 110, 200, 300, 600, 1200, 2400  
WORD\_LENGTH: 5, 6, 7, 8  
PARITY: E, O, N  
STOP\_BITS: 1, 2

Data collected  
Power on: 300,8,N,1 = 153 1001 1001  
 300,8,E,1 = 154 1001 1010  
 300,8,O,1 = 152 1001 1000  
 300,8,N,2 = 157 1001 1101  
 300,5,N,1 = 129 1000 0001  
 300,6,N,1 = 137 1000 1001  
 300,7,N,1 = 145 1001 0001  
 50,8,N,1 = 25 0001 1001  
 110,8,N,1 = 89 0101 1001  
 200,8,N,1 = 121 0111 1001  
 300,8,N,1 = 153 1001 1001  
 600,8,N,1 = 185 1011 1001  
 1200,8,N,1 = 217 1101 1001  
 2400,8,N,1 = 249 1111 1001

SETCOM Bits 1,0: Parity 00 = Odd, 01 = None, 10 = Even  
SETCOM Bit 2: Stop 0 = One, 1 = Two  
SETCOM Bits 4,3: Word Len 00 = 5, 01 = 6, 10 = 7, 11 = 8  
SETCOM Bits 7-5: Baud 000 = 50, 010 = 110, 011 = 200, 100 = 300,   
 101 = 600, 110 = 1200, 111 = 2400

SETDEV 7857   
KI = Input, DO = Print, PO = LPRINT/LLIST, CI = CLOAD, CO = CSAVE  
Command example: SETDEV KI

Data collected  
Power on: = 192 1100 0000  
 KI = 193 1100 0001 Bit 0  
 DO = 194 1100 0010 Bit 1  
 PO = 196 1100 0100 Bit 2  
 CI = 200 1100 1000 Bit 3  
 CO = 208 1101 0000 Bit 4  
 ?? = ??? 0010 0000 Bit 5  
 ?? = ??? 0100 0000 Bit 6  
 ?? = ??? 1000 0000 Bit 7

ZONE 7856 – 1-31  
Command example: ZONE 5

Data collected  
Power on: = 13  
 1 = 1  
 5 = 5  
 10 = 10  
 20 = 20  
 31 = 31

INSTAT the same?

OUTSTAT Bit Function Direction  
7-5 N/A N/A  
4 DSR IN  
3 CD IN  
2 CTS IN  
1 RTS OUT  
0 DTR OUT

USING $7895 (USING\_F/F) ; Using format, presence of decimal point, comma, etc.  
  
$7895 – USING editing character -> Bit 7: Scientific, Bit 6: asterisk fill, Bit 5: Forced sign, Bit 4: Comma separation, 01: used in checking syntax  
$7896 – USING number of characters, including sign, before decimal point  
$7897 – USING number of characters in string; $00 = unspecified  
$7898 – USING number of characters including and following decimal point  
  
X = PI Y = 1234 A$ = “ABCDEF”

Default PEEK (&7895) = 0  
BASIC Line [display ] PEEK $7895  
Print USING “###”;X [ 3] PEEK (&7895) = 1 $01  
Print USING “+###. ###”;X [ +3.141] PEEK (&7895) = 33 $21  
Print USING “###. ##^”;X [ +3.14E 00] PEEK (&7895) = 129 $81  
Print USING “###. ^”;X [ +3.E 00] PEEK (&7895) = 129 $81  
Print USING “\*######”;Y [ \*\*1234] PEEK (&7895) = 65 $41  
Print USING “\*\*\*####”;Y [ 1234] PEEK (&7895) = 65 $41  
Print USING “###,###,###”;246813 [ 246,813] PEEK (&7895) = 17 $11

CE-158 LH5811 Pin Map

|  |  |  |
| --- | --- | --- |
| Signal | Pin | Description |
| PA1 | 1 | RS232C I/F send request |
| PA2 | 2 | RS232C I/F ready to receive |
| PA3 | 3 | RS232C I/F carrier detect |
| PA4 | 4 | RS232C I/F data set ready |
| PA5 | 5 | Low battery |
| PA6 | 6 | Baud rate select |
| PA7 | 7 | Baud rate select |
| GND | 8 | Ground |
| PB0 | 9 | Centronics parallel I/F DATA 2 |
| PB1 | 10 | Centronics parallel I/F DATA 3 |
| PB2 | 11 | Centronics parallel I/F DATA 4 |
| PB3 | 12 | Centronics parallel I/F DATA 5 |
| PB4 | 13 | Centronics parallel I/F DATA 6 |
| PB5 | 14 | Centronics parallel I/F DATA 7 |
| PB6 | 15 | Centronics parallel I/F DATA 8 |
| PB7 | 16 | Centronics parallel I/F BUSY input |
| GND | 17 | Ground |
| PC0 | 18 | Baud rate select |
| PC1 | 19 | Baud rate select |
| PC2 | 20 | Baud rate select |
| PC3 | 21 | Baud rate select |
| PC4 | 22 | Baud rate select |
| PC5 | 23 | Centronics parallel I/F DATA 1 |
| PC6 | 24 | Centronics parallel I/F STROBE |
| PC7 | 25 | Centronics parallel I/F INIT |
| CS0 | 26 | VCC |
| CS1 | 27 | VCC |
| /CS2 | 28 | GND |
| RS0 | 29 | AD0 |
| RS1 | 30 | AD1 |
| RS2 | 31 | AD2 |
| RS3 | 32 | AD3 |
| R/W | 33 | Memory Read/Write |
| ME0 | 34 | ME0 Memory Designation |
| ME1 | 35 | Ground |
| W0 | 36 | Ground |
| W1 | 37 | Ground |
| GND | 38 | Ground |
| VCC | 39 | VCC |
| DME0 | 40 | Not used |
| DME1 | 41 | Not used |
| WAIT | 42 | Not used |
| INT | 43 | Interrupt request |
| RESET | 44 | Reset input |
| IRQ | 45 | Interrupt request |
| ΘOS | 46 | Internal clock in phase with LSI |
| CLI | 47 | Ground |
| SDI | 48 | Ground |
| LC | 49 | Not used |
| CL0 | 50 | Not used |
| SD0 | 51 | Not used |
| D0 | 52 | Data bus |
| D1 | 53 | Data bus |
| D2 | 54 | Data bus |
| D3 | 55 | Data bus |
| D4 | 56 | Data bus |
| D5 | 57 | Data bus |
| D6 | 58 | Data bus |
| D7 | 59 | Data bus |
| PA0 | 60 | RS232C I/F terminal ready |

System I/O Registers

PC-1500 CE-150 CE-158 Description  
F000 B000 D000 Do not use  
F001 B001 D001 Do not use  
F002 B002 D002 Do not use  
F003 B003 D003 Do not use  
F004 B004 D004 Divider Reset. Clock divider?  
F005 B005 D005 U register output.  
F006 B006 D006 Serial transfer  
F007 B007 D007 Loader Divider to F Register  
F008 B008 D008 Port C Input/ Output  
F009 B009 D009 G Register Input / Output  
F00A B00A D00A MSK Register Input / Output  
F00B B00B D00B IF Register Input / Output  
F00C B00C D00C Specify Port A I/O direction  
F00D B00D D00D Specify Port B I/O direction  
F00E B00E D00E Port A Input/ Output  
F00F B00F D00F Port B Input/ Output

PC-1500 Memory Map  
  
  
ME0 is System ROM/RAM and user RAM.  
ME1 is I/O

CE-158 subroutines/registers used in CE-158\_ROM\_HIGH

7850 - OUTSTAT\_REG  
7855 - CR/LF\_REG  
7857 - SETDEV\_VAL  
7858 - SETCOM\_REG  
7882 - STK\_PTR\_SBR-06\_SBR-32: Stack pointer for SBR 06 and SBR 32. Syntax check: Error 1 if ctrl byte in (7882)=0 and end of instruction or ": " is detected and the ctrl byte in (7882)=1  
  
7899 - VAR/ARR\_PTR\_H  
789A - VAR/ARR\_PTR\_L  
78A2 - PREV\_LINE\_H – Line number immediately proceeding  
78AE - TRACE\_ON – Address of break point line number  
78B6 - ERROR\_TOP\_H – Address of program block if error  
78B9 - ON\_ERR\_ADD\_H – Address to where program jumps on error  
79DE - UNDEF\_REG\_79DE  
79F1 - BATT/PEN\_FLAG: BATTERY FLAG or PEN CHANGE FLAG  
79FA - UNDEF\_REG\_79FA  
7A04 – If B2, 7A05 contains string. If D0, 7A05 contains variable  
7A05-7A06 - MATH\_REG\_Xa\_B6:   
7A20 – VREG: RESET Flag?  
7A22 - MATH\_REG\_Va\_B3: Error flags?  
7A26 - MATH\_REG\_Va\_B7  
7A28 - MATH\_REG\_Wa\_B1  
7BB0 - INPUT BUFFER Byte 0 START  
8307 – Seems to be start of table referenced by 8BAC  
83C0-83D0 - UNKNOWN\_TBL  
83E7 – Seems to be start of table referenced by 8B94  
B00A - DŽbut bloc program MGP 2  
CED0 – SUB13: TXFR\_RSV\_KEY: Store reserve memory start address+8 in X-Reg. Set C=0 if 2nd reserve memory area used  
D008 - SBR3A, SUB18\_9: load next character/token to U-Reg  
D00C-  
D00D-D00F,SUB19: Initialization of the program pointers  
D200-D203,SUB39: Inside BCD function w/starts @ D14F about BCD?  
DFE2 - SUB85: U-Reg = U-Reg -- X-Reg when upper limit occurs: C=0 and UH=16  
F00B – Something to do with Interrupt Flag?

CE-158 subroutines/registers used in CE-158\_ROM\_LOW

7850 - OUTSTAT\_REG  
7851 - Console 1 (RS232C)  
7852 - Console 1 (//)  
7855 - CR/LF\_REG  
7856 - ZONE\_REG  
789B - ERRORNUM  
78B4 – ERR\_LINE\_H (Line number where error is met) (H)  
79D1 - Indicates a BASIC extension table. Only strong weights (&FC) are retained. nn = Val(79D1)\*2. If Val(nn00) = &55 and the code of the word clŽ is F0 then searches for the word clŽ ˆ from nn54. OpenDV (Peripheral device select)  
  
79DE – Undefined  
7A06 - MATH\_REG\_Xa\_B7  
7A28 - MATH\_REG\_Wa\_B1: Used to poke in code snippet to switch to high bank  
7A30-7A33 – MATH\_REG\_ Sa\_B1, SREG: Loads with code snippet to switch to High Bank address 8013.  
  
D00E - Initialization of the program pointers  
DFA1 – SUB82  
DFE2, SUB85 - U-Reg = U-Reg - X-Reg when upper limit occurs: C=0 and UH=16

Does XL value passed to MAIN\_ENTRY/CLOAD signify command? Where does XH come from?  
  
Command XL  
COM\_TBL\_INPUT# ED  
COM\_TBL\_PRINT# EE  
LPTR\_TBL\_PRINT# EF  
COM\_TBL\_INIT D0  
MERGE F0

"BREAK" F0B3  
"CLOAD" F089  
"CSAVE" F095  
"COM$" E858  
"DEV$" E857  
"DTE" E884  
"ERN" F052  
"ERL" F053  
"FEED" F0B0  
"INPUT" F091  
"INSTAT" E859  
"LPRINT" F0B9  
"LLIST" F0B8  
"MERGE" F08F  
"OUTSTAT" E880  
"PRINT" F097  
"PROTOCOL" E881  
"RINKEY$" E85A  
"SETCOM" E882  
"SETDEV" E886  
"SPACE$" F061  
"TERMINAL" E883  
"TRANSMIT" E885  
"TAB" F0BB  
"ZONE" F0B4

"CONSOLE" F0B1