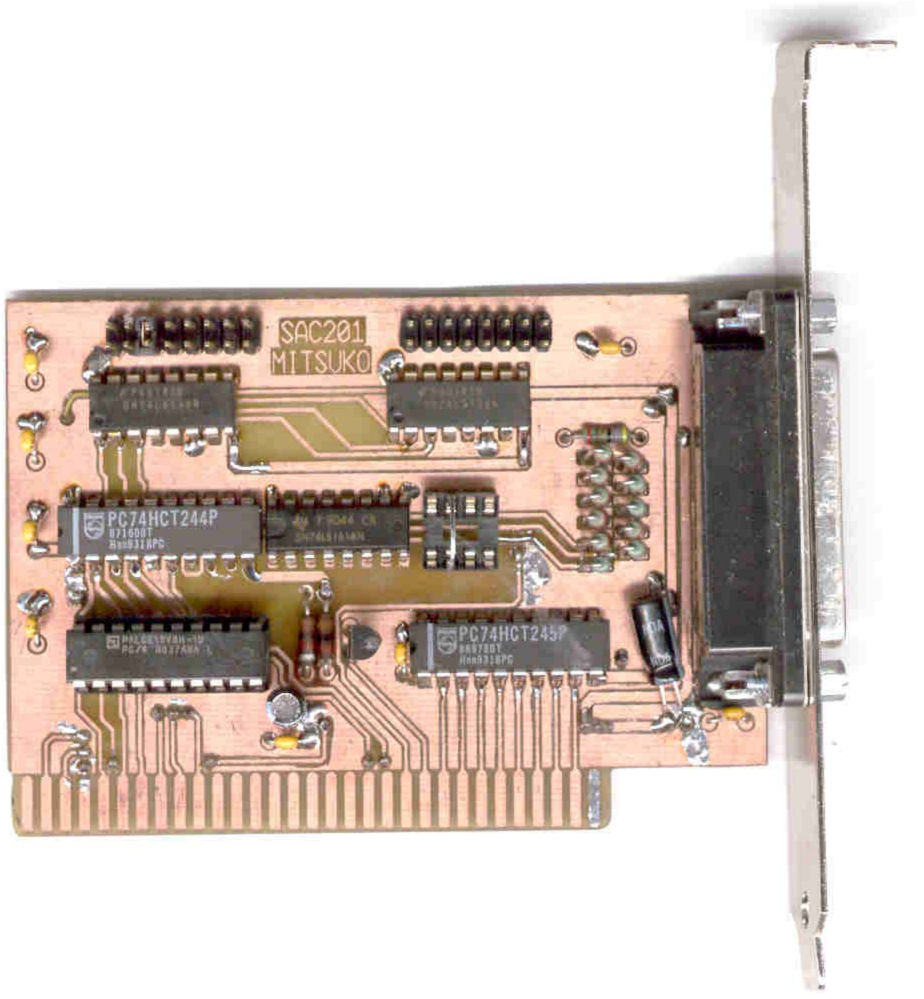
Hilosystems SAC-201 interface board for ALL-03 / ALL-03A PC-BASED universal programmer - New Software for the ALL03 - Sources

here is the diagram of the SAC-201 board and the jedec file of the PALCE16V8H U4 is provided here.



here you'll find the PCB of the SAC201 hilosystems board drawn by MITSUKO.

Bill of Materials of SAC201 board.

C2,C3,C4,C6,C7,C8,C10,C1 100nF
C5,C9 10uF
J1 IBMXT
J2,J3 HDR_16
J4 HDR_8
J5 DELTA_25HF
R1,R2 1K
R14,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13 47E
T1 2N4401
U2 74LS161A
U3 74HCT244
U4 PAL16V8
U5,U1 74LS138

U6 74HCT245

SUNSHINE EXPRO-80 / HILOSYSTEMS ALL-03A universal devices programmer and tester diagrams (409KB, 8 pages)

Hi!
If you can get a second hand Hilo Systems All03 device Programmer and tester you will be able to program with no problem the AT89CX051.
You can use with it the free software compiled with DJGPP available on this page. Also Note that such a programmer could be found for a 150 euros at some surplus or elsewhere.

Here is the last Update (20030702 , 670Kb) of the software fabio.sturman@tiscali.it . This Software includes software for programming ATMEL AT89C1051, AT89C2051 and AT89C4051 ; MICROCHIP PIC16F83, PIC16F84 , PIC16C84 ; ATMEL AT90S8515, AT90S/LS2323, AT90S/LS2343, AT89S52, AT89C51, AT89C52 , 27C512A, 27256 with the ALL03 / ALL03A programmer.

This software is compiled with DJGPP (cf. http://www.delorie.com/djgpp/) To compile it you need the following files (packages) : bnu, bsh, cpio, diff, djdev, ed, fil, find, flx, gcc, glib, gpp, grep, gtxt, mak, sed, tar, txt (look in the directory "x:\djgpp\manifest") . In the directory "c:\djgpp" create this batch file and name it "djgpp.bat" . Select the appropriate disk drive, select one of the shells uncommenting one line at the end.

```
----begin file------
@echo off
set PATH=c:\djgpp\bin;c:\djgpp\gnu\emacs\bin;%PATH%
set DJGPP=c:\djgpp\djgpp.env
set PS1=\w \$
rem uncomment the following line if you want to use a bash shell
rem bash
rem uncomment the following line if you use win-nt4-2000-xp
rem cmd
rem uncommentthe following line if you use win-95-98-me
rem command
----end file--------
```

Expand the all03-x.zip into "c:\djgpp"
Start djgpp.bat from explorer, execute the command "cd all03/commandline" and then the command "make". All the binaries and the libray will be rebuild. If you want to modify only one module, change the source for the module, save it and then execute "make" from the modules's directory

---

Previous release:

here is the good work from fabio.sturman@tiscali.it that is a new software for the ALL-03 /ALL-03A device programmer which allow to program new devices This new version allow you to program the **89c52** and **90s8515** microcontrollers. The program is still runing under MS-DOS. The implementation of a Windows driver for the SAC-201 board would be perhaps to hard for timing reasons. (Shift-Click to Download :218 Kb)

---

here is the archive of the expro60 hilo universal devices programmer which may be very similar

to the all03. This archive include the sources (equations) for the 3 PAL16R4 included in ithe expro60 named as follow : U5, U33, U35. and also diagram for the expro60. (Shift-Click to Download :780 Kb)
exp_u5.eqn (equations) , exp_u5.jed (jedec file)
this is expro60 programmer board(Motherboards) of the CHIPS (PALCE16V8H U5).

exp_u33.eqn (equations), exp_u33.jed (jedec file)
this is expro60 programmer board(Motherboards) of the CHIPS (PALCE16V8H U33).

exp_u35.eqn (equations), exp_u35.jed (jedec file)
this is expro60 programmer board(Motherboards) of the CHIPS (PALCE16V8H U35).

Diagrams of the EXPRO-60 :

Sheet 1 of 8 (EXPRO-60 INTERFACE PC)
Sheet 2 of 8 (EXPRO-60 SUB-BOARD PART I ALIM)
Sheet 3 of 8 (EXPRO-60 SUB-BOARD PART II ADRESS)
Sheet 4 of 8 (EXPRO-60 SUB-BOARD PART III LOGIC)
Sheet 5 of 8 (EXPRO-60 UP-BOARD PART I LATCH)
Sheet 6 of 8 (EXPRO-60 UP-BOARD PART II)
Sheet 7 of 8 (EXPRO-60 UP-BOARD PART III)
Sheet 8 of 8 (EXPRO-60 UP-BOARD PART IV)

here you will find the same information but for the expro80 including diagrams of the EXPRO80 and diagram and information to build a LPT1 parallel Port interface for the programmer. LINK-P1 parallel adapter for the EXPRO60-80 (ALL03 also ?) to use with the release of the Expro60-80 dated 22 August 1996. Use a standalone power power supply 220V -> +5,+12,-5,-12. to power the programmer.

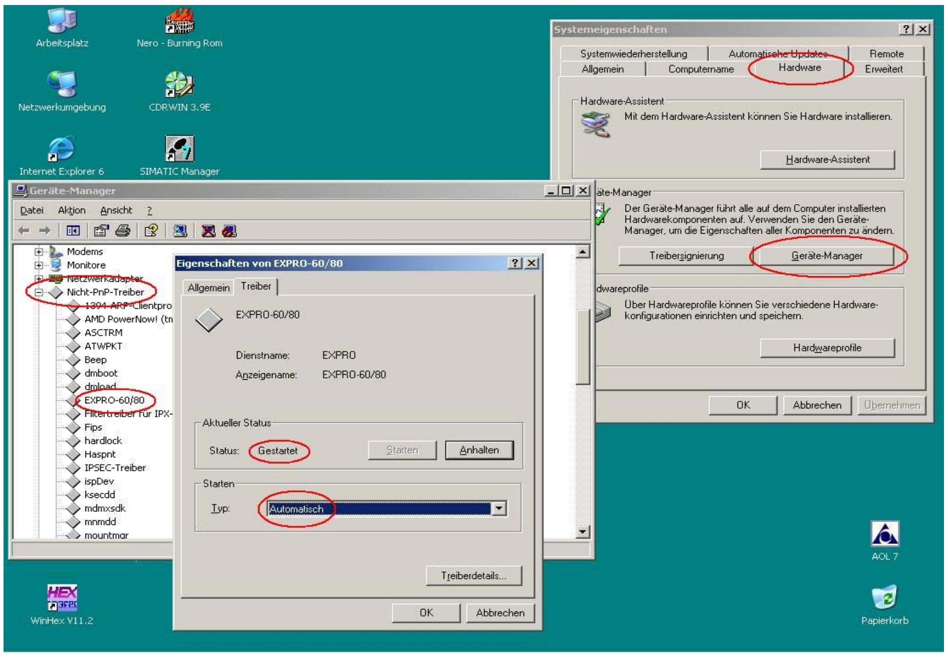
▶Sunshine Driver update disk1, disk2 .

Sunshine Expro60 / Expro80 User's Manual (72 pages 2.79Mb)

Device driver for Win NT/2K/XP written by Hubert Sack - Works only with patched software of the ALL03 or EXPRO 60/80 devices programmers & testers : (available under the terms of the Gnu Public License GPL)

Here Download the Driver **Expro.sys** in order to use an Hilosystems ALL03 / ALL03A or Sunshine Expro60 /80 with **Windows XP/2000/NT** Operating System. **(Shift-Click to download - 1.09MB)**

See **how to install the driver** with Windows XP Operating system through those shortcuts, and to verify the driver is already installed:



Systemeigenschaften

Hardware-Assistent: Mit dem Hardware-Assistent können Sie Hardware installieren.

Geräte-Manager: Der Geräte-Manager führt alle auf dem Computer installierten Hardwarekomponenten auf. Verwenden Sie den Geräte-Manager, um die Eigenschaften aller Komponenten zu ändern.

Hardwareprofile: Über Hardwareprofile können Sie verschiedene Hardwarekonfigurationen einrichten und speichern.

Eigenschaften von EXPRO-60/80

EXPRO-60/80

Dienstname: EXPRO

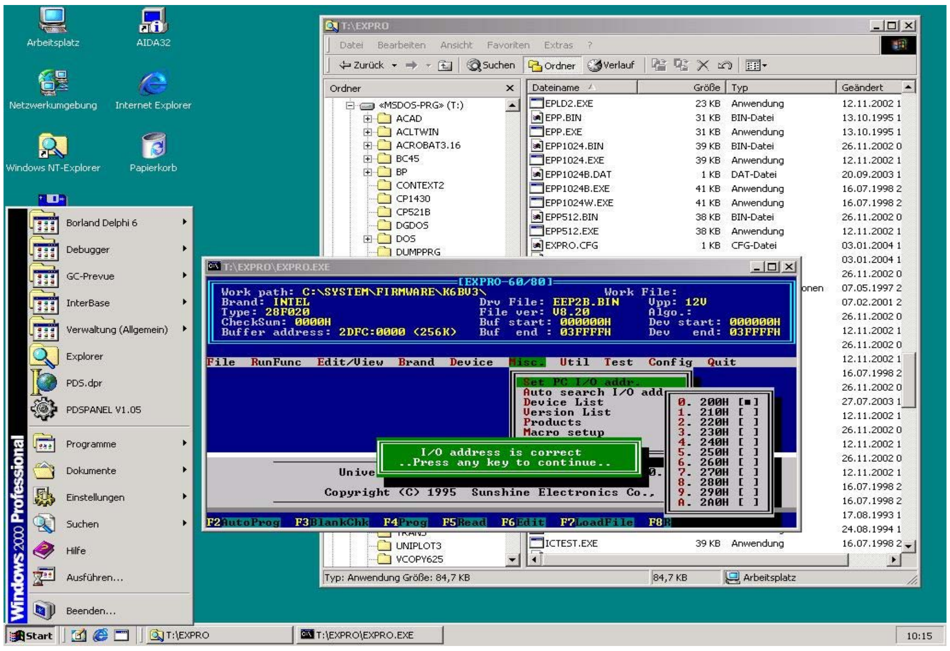
Anzeigename: EXPRO-60/80

Aktueller Status

Status: Gestartet

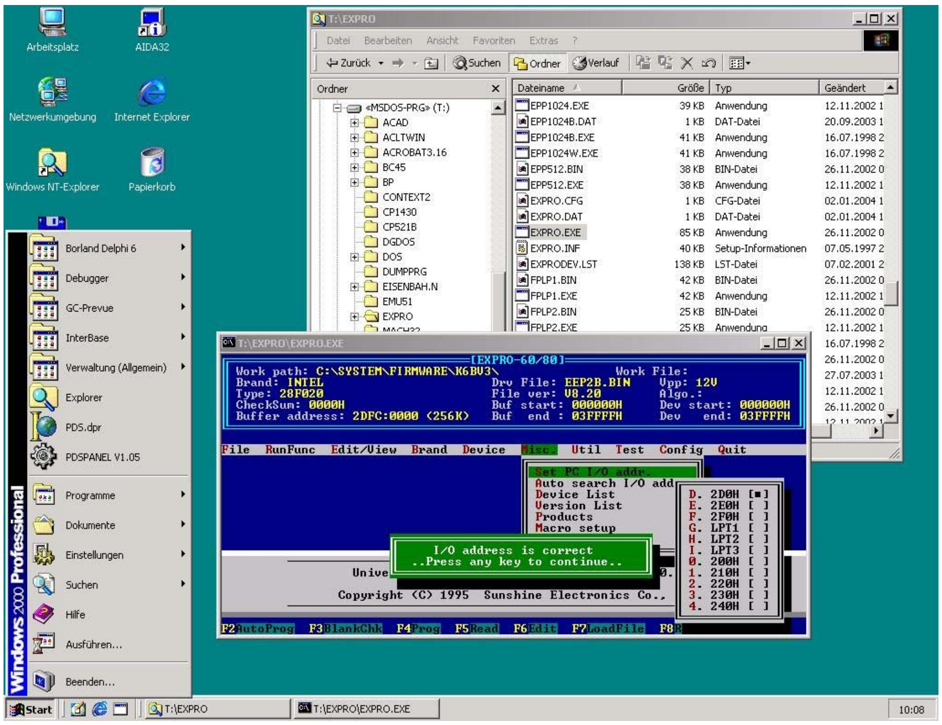
Starten

Typ: Automatisch



```
T:\EXPRO\EXPRO.EXE
[EXPRO-60/80]
Work path: C:\SYSTEM\FIRMWARE\K6BV3\                      Work File:
Brand: INTEL                          Drv File: EEP2B.BIN   Vpp: 12V
Type: 28F020                          File ver: V8.20       Algo.:
CheckSum: 0000H                       Buf start: 000000H    Dev start: 000000H
Buffer address: 2DFC:0000 (256K)      Buf  end : 03FFFFH    Dev   end: 03FFFFH

File  RunFunc  Edit/View  Brand  Device  Misc.  Util  Test  Config  Quit
                                        Set PC I/O addr.
                                        Auto search I/O add  0. 200H [■]
                                        Device List          1. 210H [ ]
                                        Version List         2. 220H [ ]
                                        Products             3. 230H [ ]
                                        Macro setup          4. 240H [ ]
                         I/O address is correct              5. 250H [ ]
                       ..Press any key to continue..         6. 260H [ ]
                 Unive                                 0.    7. 270H [ ]
                                                             8. 280H [ ]
               Copyright (C) 1995  Sunshine Electronics Co., 9. 290H [ ]
                                                             A. 2A0H [ ]
F2AutoProg  F3BlankChk  F4Prog  F5Read  F6Edit  F7LoadFile  F8R
```



Dateiname	Größe	Typ	Geändert
EPP1024.EXE	39 KB	Anwendung	12.11.2002 1
EPP1024B.DAT	1 KB	DAT-Datei	20.09.2003 1
EPP1024B.EXE	41 KB	Anwendung	16.07.1998 2
EPP1024W.EXE	41 KB	Anwendung	16.07.1998 2
EPP512.BIN	38 KB	BIN-Datei	26.11.2002 0
EPP512.EXE	38 KB	Anwendung	12.11.2002 1
EXPRO.CFG	1 KB	CFG-Datei	02.01.2004 1
EXPRO.DAT	1 KB	DAT-Datei	02.01.2004 1
EXPRO.EXE	85 KB	Anwendung	26.11.2002 0
EXPRO.INF	40 KB	Setup-Informationen	07.05.1997 2
EXPRODEV.LST	138 KB	LST-Datei	07.02.2001 2
FPLP1.BIN	42 KB	BIN-Datei	26.11.2002 0
FPLP1.EXE	42 KB	Anwendung	12.11.2002 1
FPLP2.BIN	25 KB	BIN-Datei	26.11.2002 0
FPLP2.EXE	25 KB	Anwendung	12.11.2002 1

```
T:\EXPRO\EXPRO.EXE
═══════════════════════════[EXPRO-60/80]═══════════════════════════
 Work path: C:\SYSTEM\FIRMWARE\K6BV3\              Work File:
 Brand: INTEL                    Drv File: EEP2B.BIN    Vpp: 12V
 Type: 28F020                    File ver: V8.20        Algo.:
 CheckSum: 0000H                 Buf start: 000000H     Dev start: 000000H
 Buffer address: 2DFC:0000 (256K) Buf  end : 03FFFFH    Dev   end: 03FFFFH

File  RunFunc  Edit/View  Brand  Device  Misc.  Util  Test  Config  Quit

                                   Set PC I/O addr.
                                   Auto search I/O add   D. 2D0H [■]
                                   Device List           E. 2E0H [ ]
                                   Version List          F. 2F0H [ ]
                                   Products              G. LPT1 [ ]
                                   Macro setup           H. LPT2 [ ]
                                                         I. LPT3 [ ]
                    I/O address is correct               0. 200H [ ]
                  ..Press any key to continue..          1. 210H [ ]
             Unive                              0.       2. 220H [ ]
             Copyright (C) 1995  Sunshine Electronics Co.,  3. 230H [ ]
                                                         4. 240H [ ]
F2AutoProg  F3BlankChk  F4Prog  F5Read  F6Edit  F7LoadFile  F8R
```

A "file" (it's the symbolic link to the driver) **\\.\EXPRO** must be opened in the program, which wants to do I/O on the ports. Therefore only the **patched programs** in the attachment will work.

patched program for the HILO ALL03 **(Shift-Click to download - 1.64MB)**

patched program for the Expro 60 /80 from Sunshine **(Shift-Click to download - 1.31MB)**

Thats because the I/O permission map at the end of the **TSS-descriptor** is modified by opening the driver. It's exactly the same way GIVEIO.SYS does, but not for the hole 64K I/O, because the keyboard, mouse etc. must be handled by **NT/2K/XP** like before.
With that "trick" the access to the port it as fast as under DOS because **IOPL** does not matter, there is NO interception. The ports **2E0** and **2F0** are **not supported** by the driver because they may cause a dangerous conflict with an existing COM3/4. The control to the Kernel is fully bypassed!

To install the driver just copy **EXPRO.SYS** to the **SYSTEMROOT\system32\drivers folder** and run **INSTALLDRIVER**. It installs the driver sets the startmode to AUTOMATIC and starts the driver, so there is no need to restart the PC.

A short description what the driver does (you can find it in the internet if you search for **GIVEIO.SYS** from Dale Roberts, my driver uses exacly the same "trick"):

Every task (in **NT/2K/XP** it's called **PROCESS**) have it's own **TASK STATE SEGMENT** descriptor. The **TSS** register holds the descriptor. You can see the structure of a **TASK STATE SEGMENT** descriptor in the pictures I already sent. Within the **TSS** structure is an offset-field which points to a bitmap. This bitmap is the I/O permission map. A **"0"** (bit, not byte !) allows the process to access the port. For each address there is a bit, so 64kbits for the full I/O address room is needed (8192 * 8 = 65536 = 64k). That's the reason why 2000h bytes of non cacheble memory is allocated by the driver. The bits of the addresses used for port I/O by **EXPRO60/80** or **ALL03** are cleared, all the other ones are set

to **"1"**.
The application only must open the "file". The kernel calls the "open" function of the driver.
It's executed within the context of the application process, but which CPL (current privileg level) of 0 (it's the highest). So the driver is able to "adjust" the I/O permission map of the task (e.g. process) to it's needs. Till now the application can do direct I/O calls to the port.
That's all; but it's sounds easier than it really is: To reference a new I/O permission map you must call two undocumented operating system calls.

The **patch program** does a very simple thing: It adds the necessary code the the end of the original program image and adjusts the header. So after loading the additional code is executed first, which opens the driver to support the I/O permission for the process. There are only two DOS calls used: Open file and Close file. **(Shift-Click to download - 8 KB)**

To assemble the driver you must use the **MASM32**. It's free of charge and can be downloaded for the intel web site.

The install program for the driver is compiled via Borland Delphi 3.
The patch program is compiled by using Borland Pascal 7, but it can be compiled under Turbo Pascal 6, too. Only one line must be changed:
FUNCTION uppercase (CONST instr: STRING):STRING;
must be changed to:
FUNCTION uppercase (VAR instr: STRING):STRING;
Borland Pascal 7 allows **"CONST"** parameters, the resulting code is the same as using **"VAR"**, but the compiler rejects all write references to the parameter because it's declared as read only (constant).

---

1. I/O Control Concept:

Some basic concepts need to be explained.

The main control unit is on the adapter card (SAC-201).

There are two 8-bit I/O registers on the SAC-201 that are able to control all I/O devices on the ALL-03 main module.

Register name	I/O address
IDPORT	base address+0
DATAPORT	base address+2

The base address is the I/O address of the SAC-201 as defined by DIP switches 1 and 2. These switches allow a choice of 16 possible base addresses from 200h to 2f0h (default is 2e0).

Every device on the ALL-03 main module can be controlled by writing its ID byte to the IDPORT and then reading from or writing to the DATAPORT (the ID bytes of the various devices will be shown below).

TABLE 2

Available control pins on test-socket:

1. VOPENID: 1, 5, 7, 9 to 32.

2. VHHENID: 9 to 32.

3. VCCENID: 40, 36, 34, 32, to 26, 9, 7, 5, 1.

4. VHHENCID: 32 to 28.

5. TTLID: all 40 pins can be defined as inputs or outputs.

Note: Devices that cannot use the previous pin definitions will require a special purpose external adapter.

Example: We wish to write the value 7fh to #1 D/A device (ID = e5h).

Assembly Language

```
idport equ 2e0h
dataport equ 2e2h


mov ax,0e5h ;id number setting
mov dx,idport
out dx,al
jmp $+2 ;device recovery time


mov ax,7fh ;data writing
mov dx,dataport
out dx,al
jmp $+2 ;device recovery time
```

C Language

```
#define idport 0x02e0
#define dataport 0x02e2


Outp (idport, 0xe5) ;/* ID number setting */
Outp (dataport, 0x7f) ;/* data writing */
```

---

2. ALL-03 Device ID Mapping and Definition:

The following describes each 8-bit device and its associated ID bytes:

Power Source D/A Voltage Level Setting Device:

e5h : #1 D/A named VOPID, full scale is 25.5 volt (400 ma), resolution is 0.1 volt, minimum voltage setting is 10.2 volt

Example: Writing 255 to device e5h will set the #1 D/A to 25.5 volt, writing 102 to e5h will set the D/A to 10.2 volt.

e6h #2 D/A named VHHID, full scale is 15.3 volt (400 ma) resolution is 0.06 volt, minimum voltage setting is 5.1 volt

Example: Writing 255 to device e6h will set the #2 D/A to 15.3 volt, writing 85 to e6h will set the D/A to 5.1 volt.

e7h #3 D/A named VCCID, full scale is 10.2 volt (1A) resolution is 0.04 volt, minimum voltage setting is 0 volt

Example: Writing 255 to device e7h will set the #2 D/A to 10.2 volt, writing 0 to e7h will set the D/A to 0 volt.

The levels of each power source can be set by the above three D/A devices. However, additional devices are required to apply these voltages to the specified test-socket pins.

TTL Level I/O Device:
----

e0h TTLID+0

e1h TTLID+1
e2h TTLID+2
e3h TTLID+3
e4h TTLID+4

Total 40 bits assigned to 40 pins on test-socket. LSB of e0h is assigned to pin 1, and MSB of e4h is assigned to pin 40.
Each pin can be a TTL level input or output. Before inputting from the desired pin, the user must output a high to that pin.

Example: After writing an ID byte (TTLID+i) to the IDPORT, one can then output 8 bits to the DATAPORT or input 8 bits from the DATAPORT.

VOP Level Output Control Device:

e8h VOPENID+0
e9h VOPENID+1
eah VOPENID+2
ebh VOPENID+3
ech VOPENID+4

Total 40 bits assigned to 40 pins on test-socket.
LSB of e8h is assigned to pin 1, and MSB of ech is assigned to pin 40.

Writing a high to a particular bit will apply the VOP source to the relevant test-socket pin. Writing a low will disable the VOP source output to that pin.

Example: After writing ID byte VOPENID+1 to the IDPORT, and then outputting 8 bit data to the DATAPORT, VOP will be applied to the relevant test-socket pins 9 through 16 via a 22 ohm current limiting resistor.

Note: 1. Test-socket pins 2,3,4,6,8 and pins 33 to 40 have no VOP control circuit, so VOP cannot be output to these pins
even though the relevant bit may have been set to high.

2. VOP must not be applied to any pin for more than one hour.

VHH Level Output Control Device:

f0h VHHENID+0
f1h VHHENID+1
f2h VHHENID+2
f3h VHHENID+3
f4h VHHENID+4

Total 40 bits assigned to 40 pins on test-socket.
LSB of f0h is assigned to pin 1, and MSB of f4h is assigned to pin 40.

Writing a high to a particular bit will apply the VHH source to the relevant test-socket pin. Writing a low will disable the VHH source output to that pin.

Example: After writing ID byte VHHENID+1 to the IDPORT, and then outputting 8 bit data to the DATAPORT, VHH will be applied to the relevant test-socket pins 9 through 16 via a 22 ohm current limiting resistor.

Note: 1. Test-socket pins 1 to 8 and pins 33 to 40 have no VHH control circuit, so VHH cannot be output to these pins
even though the relevant bit may have been set to high.

2. VHH must not be applied to any pin for more than one hour.

VCC Level Output Control:

edh VHHENID+0
eeh VHHENID+1

Total 16 bits assigned to 16 pins on test-socket.

edh b0 : pin 40 eeh b0 : pin 27
bl : pin 36 bl : pin 26
b2 : pin 34 b2 : pin 9
b3 : pin 32 b3 : pin 7
b4 : pin 31 b4 : pin 5
b5 : pin 30 b5 : pin 1
b6 : pin 29 b6 : not used
b7 : pin 28 b7 : not used

Writing a high to a particular bit will apply the VCC source to the relevant test-socket pin. Writing a low will disable the VCC source output to that pin.

Example: After writing ID byte VCCENID+0 to the IDPORT, and then outputting 8 bit data to the DATAPORT, VCC will be directly applied to the relevant test-socket pins 28 through 40.

VHH Level Output to Extra Pins Control Device:

The difference between these extra pins and the previously listed VHH pins is that these extra pins have no 22 ohm current limiting resistor in series with the VHH source. This is to permit some PAL VCC pins to be driven at high voltage (over 12 volts) and high current.

f5h VHHENCID+0
f6h VHHENCID+1

Total 16 bits assigned to 16 pins on test-socket.

f5h b0 : not used f6h : not used now
bl : not used
b2 : not used
b3 : pin 32
b4 : pin 31
b5 : pin 30
b6 : pin 29
b7 : pin 28

Writing a high to a particular bit will apply the VHH source to the relevant test-socket pin. Writing a low will disable the VHH source output to that pin.

Example: After writing ID byte VCCENCID+0 to the IDPORT, and then outputting 8 bit data to the DATAPORT, VHH will be directly applied to the relevant test-socket pins 28 through 32.

"Other Pins" Control Device:

Some additional pins not previously specified will be listed here.

efh OTHERENID

Total 8 bits assigned to 9 pins on test-socket

efh b0 : low will set pin 20 to ground
high will set pin 11, 30 to ground

bl : hi will output VOP (source - 2.4) volt to pin 7.
low will disable output.

b2 : hi will output VOP (source - 2.4) volt to pin 25.
low will disable output.

b3 : high will output oscillator TTL level to pins 2 and 3.

Each pin is the inverse of the other.
low will disable output.

b4 : high will output oscillator TTL level to pins 18 and 19.
Each pin is the inverse of the other.
low will disable output.

b5 : high will select 4.432 mhz oscillator
low will select 2.216 mhz oscillator

b6 : not used

b7 : not used

Note: 1. All devices are reset to a low output at PC power on.

2. Voltage drops exist between the power sources and the test
socket pins. These voltage drop factors need to be included
in the D/A power source level settings.

These voltage drops are:

VCC drop : 0.6 v, (= 15 steps)
VHH drop : 0.6 v, (= 10 steps)
VOP drop : 0.6 v, (= 6 steps)

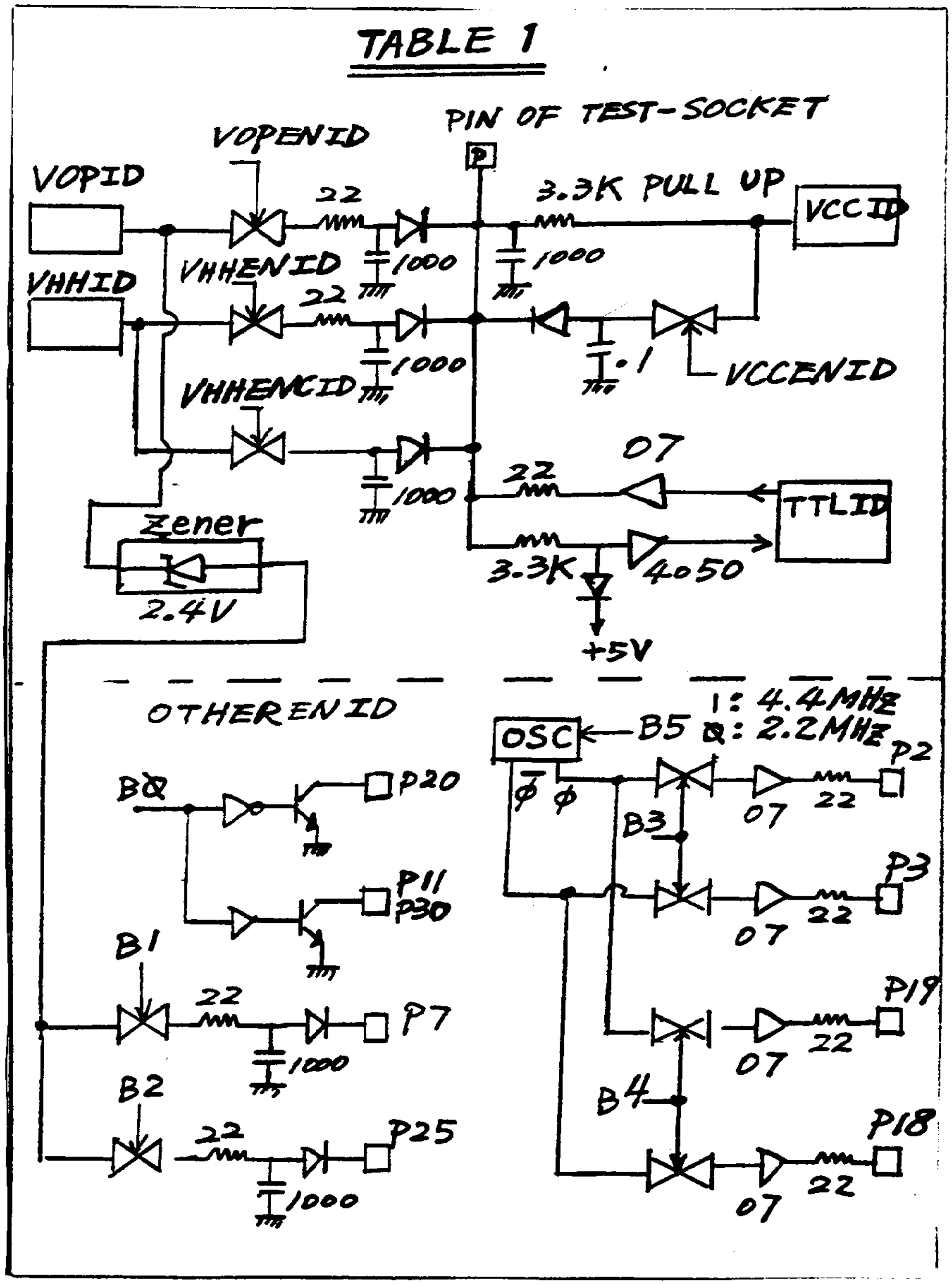
Example: The following code sets the #1 D/A to 21 volt:

```
outp (IDPORT, VOPID)
outp (DATAPORT, 210+6)
```

3. Before inputting TTL levels, or outputting VOP, VHH, VCC voltages, the user must output a high to the specified pin.
Otherwise, the relevant TTL input, or the specified supply voltage will be pulled down to ground.
This will result in damage to the affected supply.

4. In the version 1.0 of the Expro 40 we supply the source listing for TESTPRO.EXE. TESTPRO source is a good sample file for the user who wishes to develop his own software to program new devices.
here download this version of testpro with source code (this is a self-extracting archive) - Shift-Click to download (673Kb)

Source.gif scaled down to 800x600 pixels.
Right Click on your mouse and select View Image to see the original size (1371x1953 pixels)



[Back to ALL03 Resource Page](http://matthieu.benoit.free.fr/all03/)

If you have any comments or resources do not hesitate to contact me by email at matthieu.benoit@free.fr . Also If you have needs to get a SAC-201 Assembled and Fully tested board , Do not hesitate to contact me.

Email: matthieu.benoit@free.fr

Last Update : 18 décembre, 2004