

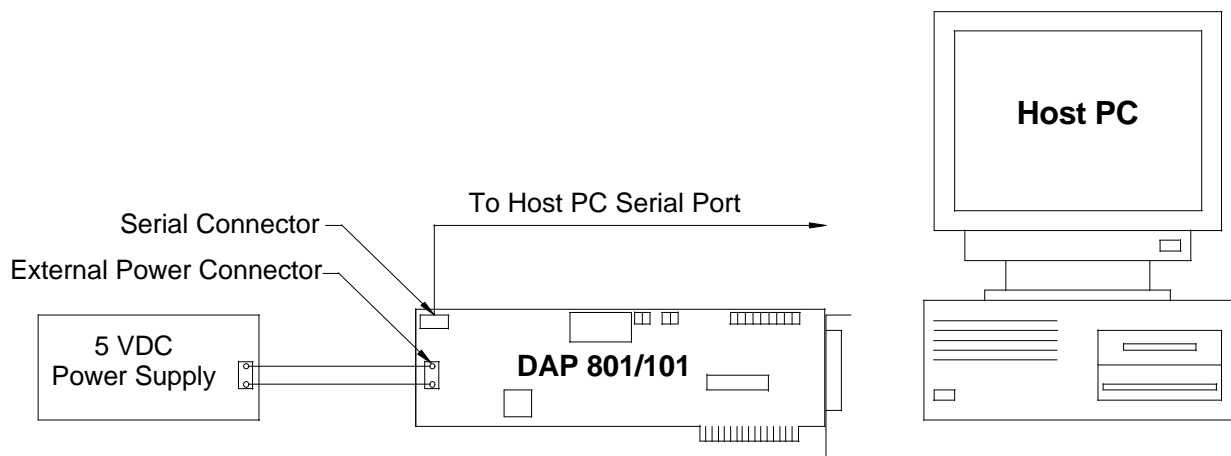
## Configuring a Data Acquisition Processor™ for Stand-Alone Operation

The nature of laboratory or industrial sites may require the operation of a Data Acquisition Processor board in stand-alone mode. Stand-alone mode can be a requirement for any data acquisition system when it is located remotely from a host computer. Applications include factory control and assembly monitoring, system warning devices, and remote data gathering stations for scientific studies.

To communicate in stand-alone mode, a Data Acquisition Processor takes advantage of the RS-232 serial interface communication hardware present on any IBM PC/XT/AT. RS-232 is a digital signal protocol that allows a PC to communicate serially with another PC or hardware device. Any Data Acquisition Processor can communicate with a PC using an RS-232 configuration; however, the DAP 801/101™ is designed especially for stand-alone applications.

### The DAP 801/101

The DAP 801/101 can be cabled directly to a serial port on the PC. The Data Acquisition Processor support software can be installed with the “serial” option so that the ACCEL device driver knows to communicate with the DAP 801/101 via the serial link. DAPview™, as well as other Data Acquisition Processor applications written for the PC, can send commands to and read text data from the Data Acquisition Processor. Figure 1 shows how the DAP 801/101 might be configured for stand-alone operation.



**Figure 1: Stand-Alone Configuration with the DAP 801/101**

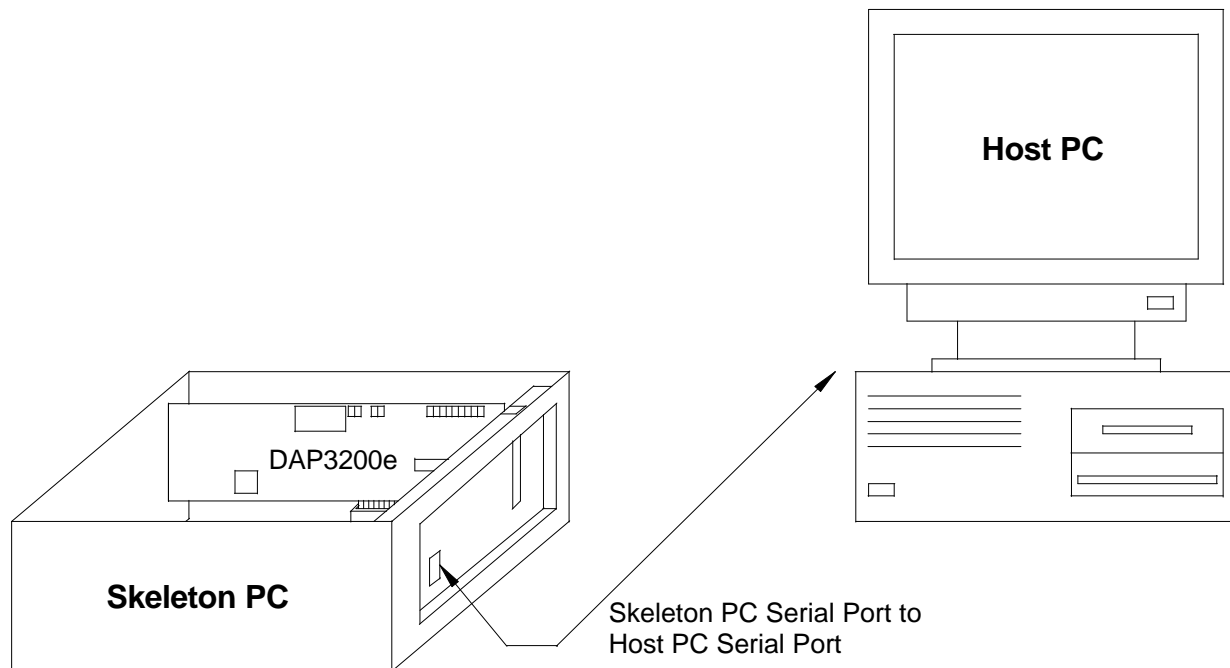
The only external equipment necessary for operating the DAP 801/101 in a stand-alone configuration is a +5 VDC power supply. Connection points for this supply are on the Data Acquisition Processor board. For communicating with the PC, cable MSCBL 013-01 attaches the serial connector on the DAP 801/101 to the serial connector on the PC. Cable lengths of up to 25 ft. are available to match

the application needs and yet still be within RS-232 specifications. Also, if normal bus operations are needed, XT edge fingers are standard on the DAP 801/101.

It may be desirable for a stand-alone application to have no connection with the PC. In this situation, the DAP 801/101 has the added benefit of loading DAPL™, the multi-tasking operating system present on all Data Acquisition Processors, from a ROM on the board. An option is also available from Microstar Laboratories where DAPL commands can be stored on an EPROM so that the DAP 801/101 can power up and begin running its application without any control or direction from a PC.

## Configuring Stand-Alone Applications Using a Skeleton PC

The DAP 801/101 is designed for stand-alone operation, but any Data Acquisition Processor may also communicate outside of the host PC. This is achieved by placing the DAP in an inexpensive “skeleton” PC. The skeleton PC does not need to have a keyboard or monitor. The DAP can communicate with the host PC via the serial port of the skeleton PC, as shown in Figure 2.



**Figure 2: Using a DAP 3200e with a Skeleton PC**

A program is needed on the skeleton PC to transfer characters received from the Data Acquisition Processor to the host PC via the serial connection, and also to receive any characters from the host PC and transfer them to the Data Acquisition Processor. The program takes advantage of the flexible and powerful “communication pipe” configuration in the ACCEL device driver. An example program, written in C, is shown at the end of this Technical Note. This short and easy-to-use program can be modified to provide a more sophisticated transfer of information with control alarms, extra processing, or even temporary storage of data in the skeleton PC's memory.

The maximum transfer rate of data over the RS-232 serial link is 9600 baud—about 100 samples (16-bit words) per second. The host computer must be configured to transmit and receive with 8 bits per character, no parity, and XON/XOFF protocol. If the standard ACCEL device driver is used, these options are set automatically.

## Using a Data Acquisition Processor board in a Laptop or Portable Computer

Although using a laptop or portable computer is not an example of a “true” stand-alone application, these PCs can fit many of the same application needs. Typically, portables require a separate battery supply to operate, while laptops use special low-power hardware and are equipped with an internal battery supply. Some portables have XT or AT bus slots, and therefore can use a Data Acquisition Processor board in the usual way. Laptops are restricted to using either RS-232, or if so equipped, a special expansion tray that allows standard bus cards to be connected. Onsite Instruments (415-964-9800) manufactures a complete workstation that uses a laptop and a Data Acquisition Processor connected via an expansion tray.

If you have any questions about using a Data Acquisition Processor in a portable or laptop, please contact Microstar Laboratories.

## Sample Program for Transferring Data

```
/* SERIAL.C                                                                    */
/* This program is designed to execute on a skeleton PC for serial              */
/* communication. It accepts characters from the host PC via a serial          */
/* link and writes these characters to the DAP. Any characters received        */
/* back from the DAP are sent to the host PC via the same serial link.       */
/* Stream I/O is used to communicate with the DAP. If a keyboard is         */
/* connected to the skeleton PC, a keystroke will terminate execution of     */
/* the program.                                                                */

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include "dapio.h"

#define MAX_CHARS 256
#define min(a,b) ((a) < (b)) ? (a) : (b)

void main ()
{
    FILE *DapIn, *DapOut, *SerIn, *SerOut;
    char ch, inchars[MAX_CHARS];
    int n,i;

    /* Open ACCEL0 (text) as a binary input file used to get data from the    */
    /* DAP using stream I/O, and as a text output file used to send          */
    /* commands and data to the DAP, again, using stream I/O.                */
    /* ACCEL6 (text) is used in the same manner but is opened as a serial    */
    /* communication pipe based on the specification in the ACOM.DAT file.    */
}
```

```

if ((DapIn = fopen("ACCEL0","rb")) == NULL) ||
    ((DapOut = fopen("ACCEL0","wt")) == NULL) ||
    ((SerIn = fopen("ACCEL6","rb")) == NULL) ||
    ((SerOut = fopen("ACCEL6","wb")) == NULL)

{
    printf("Error opening DAP device driver\n");
    exit(1);
}

fBinaryMode(DapIn);          /* Set all files to DOS binary mode */
fBinaryMode(DapOut);
fBinaryMode(SerIn);
fBinaryMode(SerOut);

setvbuf(DapOut,NULL,_IONBF,0); /* Turn off output stream buffering */
setvbuf(SerOut,NULL,_IONBF,0);

fWriteIoCtlStr(DapOut,"S,M00"); /* Set up ACCEL device driver mode */

fprintf (DapOut, "RESET\n");    /* Send a command to reset the DAP */
fflush(DapOut);                /* Flush old DAP data */

while (!kbhit())
{
    /* If characters are sent from DAP then read them and send to SERIAL */
    if ((n = fGetDapAvail(DapIn)) > 0 )
    {
        n=min(n,MAX_CHARS);
        fGetDapBuf(inchars, 1, n, DapIn);
        fPutDapBuf(inchars, 1, n, SerOut);
    }

    /* If characters are sent from SERIAL then read and send to DAP */
    if ((n = fGetDapAvail(SerIn)) > 0 )
    {
        n=min(n,MAX_CHARS);
        fGetDapBuf(inchars, 1, n, SerIn);
        fPutDapBuf(inchars, 1, n, DapOut);
    }
}

fprintf (DapOut, "STOP\n");    /* Send a command to stop the DAP */
fWriteIoCtlStr(DapOut,"R");    /* Restore ACCEL device driver mode */
fclose(DapOut);                /* Close files */
fclose(DapIn);
fclose(SerOut);
fclose(SerIn);
}

```