MODEL 134
16-BIT FAST A/D CONVERTER

LAWSON LABS, INC.
3217 Phoenixville Pike
Malvern, PA 19355

610 725-8800
or
800 321-5355

Rev. D

TABLE OF CONTENTS

INTRODUCTION

The Lawson Labs Model 134 A/D card enables an IBM PC, XT or AT computer to translate continuously variable analog voltages into their digital equivalents of ones and zeros. It also provides a means for the computer to turn external equipment on and off. These functions are the basics of monitoring and control systems. Both the analog inputs and the digital outputs are protected against overvoltages.

The Model 134 uses only four memory locations out of the 32 designated for a "prototype card." These four addresses can be changed by altering the switch settings on the A/D card. This arrangement minimizes the chance that there will be a conflict between the Model 134 and any other peripheral card.

SECTION 1. INSTALLATION

NOTE: Always handle circuit cards by the edges.

There are four small switches in the upper left corner of the Model 134. Switch #4 is unused. Place switches 1, 2 & 3 in the OFF position. This will map your A/D card into input port locations 796 through 799. If you have another special function card in your computer which uses these locations, see Section 2 for alternate mapping.

Static electricity can damage computer circuitry so care should be taken to control static discharge. Turn off the computer but do not unplug it. Remove the cover. First remove the screw which secures the bracket behind a vacant slot. Then remove the bracket. While holding the Model 134 in one hand touch the computer's chassis with the other. This procedure will harmlessly drain any static charge that may be present. Now insert the card into the connector. Press down firmly to insure that the card is seated and replace the screw in the bracket.

Attach a pair of wires to pins 6 and 13 of the mating 15-pin connector. The negative input is pin 13 and the positive input is pin 6. The other end of the wires should connect to the voltage being measured. A 1.5 volt battery is a convenient voltage source for test purposes. For a battery or other floating voltage source, a connection must be made between the negative input and ground. The computer's chassis is the best ground for most purposes.

NOTE: Strain-relief is recommended for all permanent wiring on the connector. Otherwise, physical stress may cause the failure of an electrical connection. The connector hood provided has a built-in strain-relief clamp.

FIGURE 1. CONNECTOR PINOUT

PIN 8...CHANNEL 1+
PIN 15...CHANNEL 1-
PIN 7...CHANNEL 2+
PIN 14...CHANNEL 2-
PIN 6...CHANNEL 0+
PIN 13...CHANNEL 0-
PIN 5...CHANNEL 3+
PIN 12...CHANNEL 3-
PIN 4...GROUND
PIN 11...DIGITAL INPUT
PIN 3...GUARD
PIN 10...OUTPUT C
PIN 2...OUTPUT D
PIN 9....OUTPUT B
PIN 1...OUTPUT A

Attach the connector and tighten the small screws that hold the halves together. Replace the cover. Turn on the power to the computer. If the power-on self-test (POST) does not behave normally, turn off the computer and refer to the troubleshooting section of this manual.

Connect the plus and minus wires to the voltage being measured. Make certain that the voltage is in the +/- 5 volt range. If the voltage source is floating in relation to ground, e.g. a battery, also connect the minus side of the voltage being measured to ground (not guard). For that ground connection, use the ground pin on the Model 134's connector or, preferably, chassis ground at the computer. If the computer chassis is not accessable, use the ground pin of the grounded wall outlet where the computer is plugged in. Remember that the positive and negative inputs on the Analog Interface should both stay within 6 volts of ground potential. See FIGURE 4.

NOTE: For maximum reliability, connect all unused inputs to ground.

SECTION 2. PERFORMING A/D CONVERSIONS

In the absence of other instructions, the A/D card will measure the voltage at input channel #0. Because the wires that you attached are connected to input #0, we need not concern ourselves with the channel selection process at this time. In essence, one INPut instruction begins the conversion. 72 microseconds later the answer is stored on the Model 134 where it remains until it is read with two INPut instructions, one for each byte of the result.

The Model 134 uses four port addresses. Writing to any of those addresses sends information to the card. Three of the addresses are for reading information from the card. If your computer contains other special-purpose cards, the possibility of port address conflict arises. Compare FIGURE 2 with the documentation of other special-purpose cards present and select switch settings that avoid port conflicts.

FIGURE 2. DIP SWITCH SETTINGS

SWITCH POSITIONS			ADC	ADDRESSES
#1	#2	#3	ADDRESS	USED
ON	ON	ON	769	768-771
OFF	ON	ON	773	772-775
ON	OFF	ON	777	776-779
OFF	OFF	ON	781	780-783
ON	ON	OFF	785	784-787
OFF	ON	OFF	789	788-791
ON	OFF	OFF	793	792-795
OFF	OFF	OFF	797	796-799

An OUT statement to any of the four addresses in the range is equivalent. Reading the lowest port in the range provides status information. Reading the next address (ADC) yields the less significant byte of the answer. Read the third address for the more significant byte of the answer. Reading the highest port address has no effect. Change the ADC (Analog to Digital Converter) address in line 10 of the following BASIC program if an alternate switch setting is used.

```
10 ADC = 797
20 X = INP(ADC): REM start conversion
30 COUNT = INP(ADC + 1) * 256 + INP(ADC)
40 VOLTS = (COUNT - 32768) * 1.52588E-04: REM scale
   in volts
50 LOCATE1,1: PRINT USING "##.####"; VOLTS
60 GOTO 30
```

Line 10 assigns a variable name to the port address. Line 20 begins the conversion. Line 30 reads the raw count which is an integer between 0 and 65535. Notice that the more significant byte of the answer (ADC + 1) must be read first because reading the low byte starts a new conversion. Line 50 displays the result and line 60 repeats the process.

RUN the program. (Use Ctrl Break to exit.) The voltage should appear in the upper right-hand corner of the screen. The Model 134 Analog Interface resolves 0.153 millivolts. A fresh D cell should read about 1.5 volts. With reasonably short wires, or with shielded wire, expect to see about +/- 1 count (.153 millivolts) of scatter. There is virtually no filtration in the Model 134 so any noise on the input will appear as variability in the readings. Connecting both of the input wires to ground will cause a potential of zero volts. The reading with the input shorted should be within a millivolt of zero (32768 counts). It is normal to see slightly increased scatter at zero volts. An open circuit will read unpredictably. An overvoltage will read 4.9999 or -5.0000. The input channels are protected against transient overvoltages of up to 125 volts and continuous overvoltages of 40 volts. Reverse the wires and note the polarity change. The plus sign is implicit and does not appear on the screen.

The conversion itself takes 72 microseconds. Reading the low byte of the answer also initiates a conversion. In the above program example, line 20 need only execute once because a new conversion begins each time the low byte is read in line 30. In BASIC, and in all but the fastest computer languages, the conversion will be over before the answer can be read. Routines that take advantage of the Model 134's speed must be written in fast languages. Those routines poll a status byte at address ADC-1 to determine when the conversion has been completed.

SECTION 3. SELECTING OTHER INPUT CHANNELS

First connect the other voltage to be measured to an unused pair of input pins on the Model 134's connector. See Figure 1. The four available channels are numbered 0 through 3. To select a channel, substitute the channel number to be selected for the "N" in this instruction.

```
OUT ADC, 16 * N
```

Channel 3 could be selected using the instruction OUT ADC, 48 or OUT ADC, 16 * 3.

Here is a sample routine which will measure the voltage at each channel and print the data. It is also on the support disk called DEMO134.BAS.

```
10 KEY OFF
20 CLS
30 INPUT "Enter ADC (check your manual) "; ADC
40 IF ADC < 769 or ADC >979 THEN PRINT
   "769 <= ADC <= 797": GOTO 30
50 STATUS = ADC - 1
60 CLS
70 LOCATE 24,30: PRINT "Press Esc to exit"
80 LOCATE 5,1
90 FOR CHANNEL = 0 TO 3
100 OUT ADC, CHANNEL * 16: REM select channel
110 X = INP(ADC): REM start conversion
120 COUNT = INP(ADC + 1) * 256 + INP(ADC): REM read card
130 VOLTS = (COUNT - 32768) * 1.52588E-04: REM scale
140 PRINT "Channel "; CHANNEL; " counts = ";
150 PRINT USING "#####"; COUNT;
160 PRINT, "voltage =";
170 PRINT USING "##.####"; VOLTS
180 NEXT CHANNEL
190 PRINT: PRINT "Digital input is ";
200 IF INP(STATUS) AND 128 THEN PRINT "OFF" ELSE PRINT "ON "
210 GOTO 80
```

Running the DEMO134 program with no connections made on channels 1 through 3 will do no harm. However, the data from those channels will be meaningless. Make a back up copy of the support disk. Start BASIC or BASICA and put the working copy of the support disk in drive A and type RUN "A:DEMO134". Provide the ADC address when prompted. You will see the raw count and voltage at each of the channels displayed on the screen. The current status of the digital input is also displayed. See the next section for a detailed description of the digital input.

The FOR NEXT loop from line 90 to line 180 steps through the channels and displays the readings. Notice that the conversion is explicitly started in line 110. This step makes it clear that the answer obtained in line 120 came from the input channel selected in line 100. Notice that after the program has run, channel 3 is still selected. Turning the power off and on or OUT ADC, 0 will reselect input channel zero.

SECTION 4. USING THE CONTROL OUTPUTS & DIGITAL INPUT

There are four optically isolated output lines on the A/D card. These are intended primarily for expanding the number of input channels using an external multiplexer. Any channel can be expanded to 16 channels with an external multiplexer. The optically isolated control lines prevent ground loops from being formed. Configurations of up to 64 analog inputs are possible with four Model 17 Multiplexers. The outputs can also be used for digital control. Think of each one as a contact closure to guard.

The output port has the same address as the input channel selector. The lower four bits appear, optically isolated, at the connector. A number from 0 to 15 sent to port ADC will appear on the output lines, coded in binary. The instruction OUT ADC, 3 will cause outputs A and B to become active. Below is a list of codes and the corresponding active outputs. Note that the sequence is that of counting in binary. 1=active.

FIGURE 3. OUTPUT CODES

CODE	OUTPUT			
	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

If it is necessary to alter the output code without affecting the input channel selection, the input selection code should be added to the output code. For example, to select input channel 1 with output code 15 use OUT ADC, 16 + 15 or OUT ADC, 31.

Electrically, each output line is the collector of an optically coupled NPN transistor whose emitter is connected to guard. Note that guard is not connected to ground on the Model 134. The guard is usually connected to the external device's ground. Each output line can drive two low-power TTL loads or many CMOS loads. An external pull-up resistor (from the collector to the external positive supply) must be used. Typically, 20 K ohms for low-power TTL or 47 K ohms for CMOS is appropriate. When the output code is 0 all four phototransistors are off and no current is drawn through the transistors. If the input to an inverting buffer is connected to output line A (with the appropriate pull-up resistor), the input level is forced high by the resistor so the buffer output would be 0. If the output code is 1, the A transistor will turn on and pull the buffer input to a logic level 0. The buffer output will then be a logic level 1.

The digital input can be a contact closure to ground or a TTL or 5-volt CMOS logic level, active low. This line reads the digital input.

```
     IF INP(ADC-1) < 128 THEN PRINT "ON " ELSE PRINT "OFF"
```

SECTION 5. USING THE FAST SAMPLING SOFTWARE

The FASTSAMP software is unprotected and fully listable. It can function as a self-contained data acquisition system or it can be included, all or in part, in your BASIC programs. FASTSAMP is written in BASIC and machine language. Most of the BASIC code is the input section which allows you to define a sample run. That definition is contained in an integer array called SAMPLE%. The array is dimensioned at 84 elements. The machine code occupies the top 768 bytes of BASIC'S 64K address space. The address of SAMPLE%(1) is passed to the machine code in a CALL instruction and the specified pattern of data points are stored in high memory. The data can be moved to a disk as a sequential text file suitable for importing by Lotus 1 2 3 (reg. trademark) or can be saved as a binary image using the BSAVE command.

Load BASIC or BASICA and type RUN "FASTSAMP". FASTSAMP begins by loading two binary files from the disk, SAMP and SAMARRAY. FASTSAMP will look only in the default directory for those files. SAMP is the machine code itself and SAMARRAY is a bit image of the SAMPLE% array. The last SAMPLE% array that was saved on the disk will be the starting point for the session. FASTSAMP does not do comprehensive error trapping so double check your entries. If you mis-enter something, keep going and you will be given a chance to correct the mistake. You will be prompted for the following information:

1) IN FASTSAMP, sampling runs are timed and fine-tuned for accurate sampling rates. Under most circumstances, 1% timing accuracy can be obtained. If you are running FASTSAMP for the first time on a particular machine, or if the clock rate has been changed, then select T for new timing. When you select new timing, the length of one interchannel delay count is established. You will normally press space to continue. Then, the last calibration (stored in the SAMARRAY file) is used for timing the interchannel delay.

2) Enter ADC (see Section 2).

3) The Model 134 can run at double speed at half the resolution. Select 8- or 16-bit resolution.

4) The interrupts may be disabled during the sample run. Interrupts can cause irregularities in the sampling interval. If the interrupts are disabled functions running in the background such as ON TIMER GOTO will be disrupted. Also, the time of day clock doesn't run when the interrupts are off.

5) The starting address for the memory data table has to be entered next. You need at least 128K of main memory and you must know if anything is residing in high memory, a virtual disk program, for example. Also be aware that some DOS commands like MODE and PRINT cause DOS to allocate more memory. You should be able to determine where the beginning of free RAM is by loading BASIC and typing

```
   DEF SEG = 0: X = PEEK(&H511) * 256 + PEEK(&H510):
   PRINT X * 16 + 65536: DEF SEG
```

The first free address above BASIC's 64K space will be printed. Not all BASICs update these memory locations. You may need to use other methods to determine where free memory begins in your machine. The upper limit for the data table will be the amount of installed memory up to 655360 (640K). That means that sampling runs of 270K 16-bit data points can be accommodated. The starting memory address for the data table is always a multiple of 16. If you enter a number that isn't a multiple of 16 the remainder will be discarded.

FASTSAMP does not check to see if the data will overwrite BASIC or portions of the operating system. If your data table includes non-existent RAM, no harm will be done but the data stored there will be irretrievable.

6) For multiple channel routines a brief delay after switching channels may be desirable. This pause allows very high impedance voltages to settle to 16-bit accuracy. If you don't need all the available speed, a delay count of 1 is recommended. If you are using an external multiplexer, a delay of at least 0.5 milliseconds will be needed for maximum accuracy. Delay counts above 32767 will cause an OUT OF RANGE error.

7) Up to 256 samples can be averaged to obtain each data point. This feature is useful for compressing slowly changing data and for noise suppression. Note that very noisy, highly averaged data, even if it is reproducible, may suffer decreased accuracy. The number of samples to average must be a power of two. If a number besides 1, 2, 4, 8, etc. is entered, a power of two will be proposed.

8) From 1 to 64 data points can be taken at each scan. The data points can be taken in any sequence. For example, a scan could consist of channel 3 being sampled 63 times followed by channel 0 being sampled once. At this point, enter only the number of points per scan desired. After the number of data points per scan has been entered, if anything has been changed (or if you just started FASTSAMP) there will be a pause of about 15 seconds while timing is determined. If one scan per second or faster cannot be achieved, you will be asked to reduce the averaging or interchannel delay.

9) Enter the scanning rate in scans/second. Rates from .5 to 22,000 scans/second are possible. Normally rates slower than one scan/second can be programmed directly (or use Lawson Labs PC64 software). If you enter a rate faster than the maximum attainable for the sampling run you have defined, the highest allowed rate will be displayed. FASTSAMP will change the rate you enter to the nearest nominally available rate. Remember that interchannel delays and averaging will lower the maximum scanning rate.

10) Next enter the number of scans per sample run. If you're not sure there is sufficient memory, proceed and let the program calculate the end of the data table for you. After the number of scans is entered, a review of the sampling run is presented. If everything is correct answer "Y" to the prompt. If you answer "N" another pass through everything is offered. Take particular note of the data table start and end addresses. Make sure the end address is not above 655360. Also note the sample run duration so that you don't give up too soon on a long sampling run.

11) After the sample run has been defined, you will be given the option of verifying the timing. If anything bearing on the timing of the sampling run has been changed, the timing should be verified. If the sampling rate selected cannot be achieved within 1%, then a possible rate will be proposed. Otherwise, the timing will be fine-tuned and you will proceed.

12) If you want to load a previously BSAVEd data file answer "Y" to the prompt. Note that the file to be loaded should have the same number of points per scan and the same number of scans as the current configuration. You will be prompted for the file name. Skip to paragraph 14.

13) Finally, the control code for each data point must be entered (see Sections 3 & 4). If no external multiplexer or control device is in use, then codes 0, 16, 32, & 48 will select the four channels. If a Model 17B multiplexer were connected to channel 2, then control codes 32 through 47 would select the 16 multiplexed channels. FASTSAMP displays each control code in turn. If a proposed control code is correct, just press RETURN to leave it unchanged. You will be given a chance to review or revise the control codes after the whole scan has been defined. Holding down the RETURN key will speed you through the review.

14) If you don't want the sampling routine to execute, Press the Esc key at the prompt. Otherwise press SPACE to begin sampling. You may want to save your sampling run definition on the disk before attempting to execute it. To do that, press Esc now and skip to 16.

15) If the "Sampling complete" message does not appear after the sampling duration has elapsed, you have probably overwritten some vital memory contents. If Ctrl-Alt-Del does not answer, you will need to resort to the power switch or reset button.

16) If you want your data saved as a text file answer "Y" to the prompt and provide the full file name specification including the drive designation and file name extension, if any. If you want to import your data into Lotus 1 2 3 (reg. trademark) you will need to use a .PRN extension. The text file is a standard ASCII sequential file with one entry for each data point and a new line for each scan. Versions vary, but Lotus may allow as few as 34 data points/scan and as few as 2048 scans/data run. The text files can also be read from a separate program or you can use the DOS TYPE command to review the data. With the DOS prompt, typing

```
          TYPE filename > LPT1
```

will send the whole file to your printer.

17) The raw data will now begin scrolling down the screen. Remember that 0 counts = -5 volts, 32768 counts = 0 volts and 65535 counts = 4.9999 volts. Press SPACE to pause or to skip the rest of the data. Press SPACE again to continue scrolling or select the exit or the restart option.

18) You will be offered a chance to save the sampling run definition on the disk. If you save it, the previous definition will be overwritten and the next time you run FASTSAMP you will start from the current status. If you don't save it, once you exit the program your current definition will be gone.

19) Select the restart option to take more data or to change the sample run. If you want to save the data as a binary file, you should exit now and issue the BSAVE command(s) from immediate mode. To BSAVE your data, first type DEF SEG = X/16 where X is the start of your data table. Then type BSAVE filename, 0, TABLELENGTH. Use DEF SEG to restore the default data segment. If your table is more than 64K long you will need to save it in blocks of 64K or less. Use 65536 instead of TABLELENGTH in the first BSAVE then type DEF SEG = X/16 + 4096. Repeat the BSAVE changing the filename to indicate part 2. The third DEF SEG would be X/16 + 2 * 4096. Continue until all the data is saved.

If you want to incorporate a sampling run into a BASIC program of your own, it isn't necessary to include all of FASTSAMP in your program. Instead you can generate and save the SAMPLE% array using FASTSAMP, then BLOAD that array and the machine language driver into your application program. The uses for all the elements in the SAMPLE% array are described in the FASTSAMP program listing. There should be enough information there for you to make small changes in the sample run by changing the SAMPLE% array in your application program. You may want to use the digital input or an extra input channel as a trigger for synchronizing the sampling run.

There are several things to remember when incorporating FASTSAMP into your program. Start your program with these lines:

```
  10 CLEAR , &HFD00
  20 DEF SEG
  30 BLOAD "SAMP", &HFD00
  40 DIM SAMPLE%(84)
  50 BLOAD "SAMARRAY", VARPTR(SAMPLE%(1))
  60 SBRT = &HFD00
```

The CLEAR statement must set aside at least 768 bytes for the driver. The first BLOAD puts the machine language driver in that reserved space. After the SAMPLE% array has been dimensioned the second BLOAD assigns the values to the array that had been previously assigned and saved by FASTSAMP. Note that those files are assumed to be present on drive A. To begin a sampling run, start with a DEF SEG to make certain that the default segment is current, then use

```
       CALL SBRT (SAMPLE%(1))
```

to begin sampling. The data can be retrieved by setting DEF SEG = X/16 (where X is the start of the data table) and PEEKING at the data. The first answer would be PEEK(0) + PEEK(1) * 256 for a 16-bit routine.

SECTION 6.   USING MACHINE LANGUAGE

Using the Model 134 from machine language requires checking the status byte to determine if the conversion is complete. The least significant bit of the status byte is set to one by a start conversion and resets to zero after the high byte of the answer is ready (36 microseconds after the start of the conversion). The high byte can be read while the low byte is in progress without disrupting the conversion. The next least significant bit of the status byte also goes high when the conversion starts. That bit is reset when the entire conversion is complete.

Assume the DX register contains ADC. The normal sequence to read the card would be:

```
      IN AL, DX     ;Start conversion
      DEC DX        ;Point DX at status
TST:
      IN AL, DX     ;Read status
      AND AL, 02    ;Mask conversion complete bit
      JNZ TST       ;Wait for complete conversion
      INC DX        ;Point DX at ADC
      IN AX, DX     ;Read both bytes
```

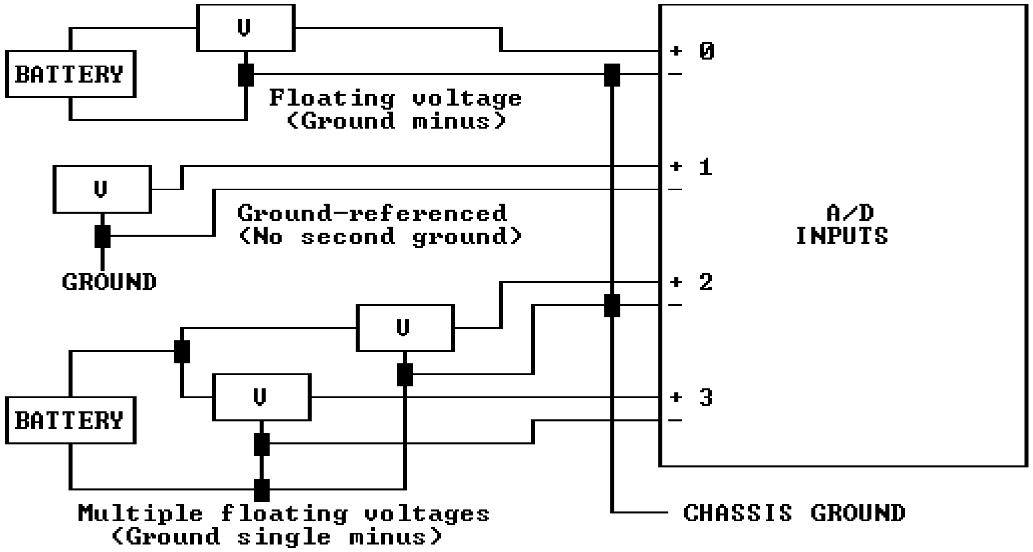
Remember that the IN AX,DX starts a new conversion. If you want that next conversion to be at a different input channel, the new control code should be output just after the conversion completes.

For 8-bit resolution, the AND statement should read AND AL, 01. The 8-bit answer will then be loaded into the AH register. The conversion started by the IN AX, DX command will yield a valid answer even though the previous conversion was still in progress when the start conversion was issued.

SECTION 7. OPERATING SUGGESTIONS

The Model 134 A/D card features fully differential inputs. A basic understanding of differential measurements will help you to use your card to best advantage. The plus and minus input pins should be wired directly to the voltage being measured. In this way, it is assured that the only current flowing in the wires will be the input current of the A/D converter. Because the wires have finite resistance, any current flowing will cause a voltage drop and a corresponding error. The A/D card requires a vanishingly small input current so the error caused by even very long wires is negligible. For proper operation of the Analog Interface it is necessary that the positive and negative inputs both be within 5 volts of ground. For a floating voltage source, this is generally accomplished by connecting a third wire between the computer chassis and a ground terminal at the source of the measured voltage. Ground currents may flow in this wire, but the resulting voltage drop will not cause a measurement error. If the source of the voltage being measured is a line-connected device with a three-pronged plug, its output voltage may or may not be referenced to ground. In that case, try with and without a ground connection for best performance. Redundant grounding can cause ground loops. Ground loops can cause unpredictable behavior.

FIGURE 4. TYPICAL INPUT CONNECTIONS



The Model 134 filters very high frequency noise only. Any noise below 1 megahertz will cause increased scatter. If your voltage source is noisy, scatter will result. Best results are obtained with filtered, buffered voltages. Electrical noise travels through the air and can be picked up by interconnecting wires. The first defense against noise is shielding. Use shielded wire with the shield connected at one or both ends to ground. (See above). The lower the impedance of the voltage source, the less susceptible the wiring will be to electrical noise. If noise problems persist, try to locate the source of the interference and shield it. Electric motors, electric heaters and flickering fluorescent lights are potential sources of interference.

SECTION 8. TROUBLESHOOTING

GENERAL:

If the A/D converter appears not to be working, first check all external wiring and all port addresses. Double check the switch settings on the Model 134. Be sure that all input voltages are within range. If another special function card is present, check for conflicting port addresses. Also, try adding a wire from the negative input to chassis ground. Make certain you are not using guard instead of ground.

1: Results seem to be meaningless numbers

The most likely culprit is a loose wire or an input with nothing attached. Try connecting the + and - inputs to ground. If the voltage does not read close to 0, the problem is in the wires themselves or in the channel selection. Check that the program has been correctly entered.

2: The answers are consistent but wrong

The ADC port address is wrong or in conflict with another card. Also, make sure all inputs are within range.

3: Readings show excessive scatter

First check for proper grounding. If there is no connection between the negative input and ground, try one. If there is such a connection, try removing it.

If the card reads a 1.5 volt battery properly but still returns widely varying numbers when measuring a voltage, the voltage could be oscillating or noisy. If you check the voltage with another voltage-measuring device remember that some meters will average out oscillations or noise to produce a steady reading. Try switching to an alternating current scale. If you get a reading the voltage is certainly unsteady. The A/D card will filter out noise above a megahertz but any noise below those frequencies will cause increased scatter.

SECTION 9. INTERNAL ADJUSTMENTS

The two adjustments on the Model 134 are gain and offset controls. The gain adjustment varies the reference voltage used during the A/D conversion. The long-term stability of the initial calibration is excellent. If you decide to alter the original gain setting, be sure your voltage reference or test meter is sufficiently accurate and has been calibrated recently. Remember that if you change the gain setting you may not be able to restore the converter to its original accuracy.

The adjustments are at the top edge of the A/D card at the rear. The potentiometer nearer the front of the computer is the offset adjustment. It is best to make any adjustments with the card at its normal operating temperature. First set the offset for zero volts with both inputs connected to ground. Then adjust the gain to match a known voltage.

TYPE:	successive approximation
INPUT RANGE:	+4.75 to -5 volts guaranteed +/-5 volts typical
RESOLUTION:	0.153 millivolts
ACCURACY:	0.03% of full scale (0.01% available)
INPUT IMPEDANCE:	10000 megohms
CONVERSION TIME:	72 microseconds at 16-bit resolution or 36 microseconds at 8-bit resolution
MAXIMUM SAMPLE RATE:	12,000 samples/second at 16-bit resolution 22,000 samples/second at 8-bit resolution (using included software)
ANALOG INPUTS:	4 channels, fully differential protected up to 125 volts transient or 40 volts continuous
DC COMMON MODE RANGE:	+/- 6 volts
DIGITAL INPUT:	single-bit, TTL or 5 volt CMOS compatible
DIGITAL OUTPUTS:	4 latched optically isolated lines protected to 200 volts each output will drive two low-power TTL loads or many CMOS loads
TYPICAL POWER CONSUMPTION:	+5v.......210 milliamps +12v.......24 milliamps (after warmup) -12v.......18 milliamps
SIZE:	3.9 x 7.7 inches 0.625 inches maximum height

LIMITED WARRANTY

The Lawson Labs, Inc. Model 134 Analog Interface is guaranteed against defects in materials and workmanship for a period of one year from the date of delivery. Products must be returned to Lawson Labs for warranty service. Contact Lawson Labs at 800 321-5355 for return authorization before returning anything for service.

The above warranty is in lieu of all warranties express or implied. Lawson Labs will not be liable for indirect or consequential damages caused by any defect in this product. Some states do not allow the limitation of consequential damages, so the above exclusion may not apply to you.