Construction Project:

COMPACT, LOW COST PC-BASED DAQ MODULE

Here's the design for a very compact and low cost module which allow virtually any PC to be used for quick and easy data acquisition and control. It connects to any standard parallel printer port, and despite its tiny size provides eight analog inputs, four digital inputs and four digital outputs — all controlled from the PC using matching software.

by PETER SIMMONDS

Martin Luther King once said "I have a dream", but his dream and mine were markedly different. *My* dream was to produce a data acquisition and control unit for an IBM compatible PC. It had to be so small that I could carry it around in my pocket, like one of those dongles you put on the back of a PC. It had to use the serial port or the parallel port, so I could use it on my desktop or my portable PC without having to open the computer's case.

It was also to be manufactured using normal dual-in-line ICs, not surface mounted IC's, and it would be powered from the PC. The ADC used in it had to have 12-bit resolution, more than one analog input — and what's more, I wanted to have both digital inputs and digital outputs.

'In your dreams', I was told, but after some agonising I have made that dream come true. Well, almost true: I couldn't power it from the PC. Although it is externally powered the unit still has very low power requirements and can be powered by a plug pack or a 9V to 12V battery, thus making it suitable for portable situations.

I started off with one of those DB25 'Gender Changer' hoods, which I had picked up from a Jaycar store some months before. Its small size and price made it perfect for the job, but it meant every component used had to earn its place or it was out. There would be very little 'fat' in this design. Also all external digital and analog I/O signals had to come in through a DB25 connector.

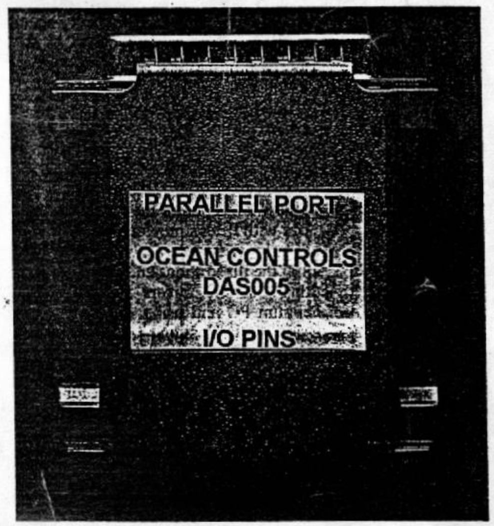
After a bit of brain rattling, the serial port on the PC was ruled out because I would have had to use a small microprocessor. So that left only the parallel port.

I had liked the look of the Maxim MAX186 when it was released by Maxim a couple of years ago. It was slimmer, and had more slender and elegant legs than the others. (Oops, I must stop breathing in those solder fumes!) What made this ADC so suitable was that it operated using a serial bus instead of a parallel bus. This meant less space was required on the PCB because of its smaller size — there wouldn't be so many tracks and not so many lines from the parallel port would be tied up controlling it. It also had a fistful of features, such as:

- An internal analog multiplexer that could be configured for eight single-ended inputs or four differential inputs.
- An internal voltage reference;
- A read input voltage range of 0 - 4.096V;
- 12-bit resolution;
- A conversion time under 10us (it's no slacker);
- A CMOS design with low power requirements;
- It requires a only a few external capacitors to filter noise from the power supply and provide compensation; and
- It was economically priced.

The only extra components I had to add was a resistor on each analog input to protect it against current overload, if a high voltage was inadvertently connected to it.

The next decision was to use a 74HCT373 for the digital I/O. This 8-bit latch is available everywhere, its cheap and it is able to sink and source a reasonable amount of current. I could use four of the latches for the inputs and the other four for the outputs.



The author's prototype for the DAQ module, shown here a little larger than actual size. It's built inside a DB-25 'gender bender' backshell, yet provides no less than eight analog inputs, four digital inputs and four digital outputs.

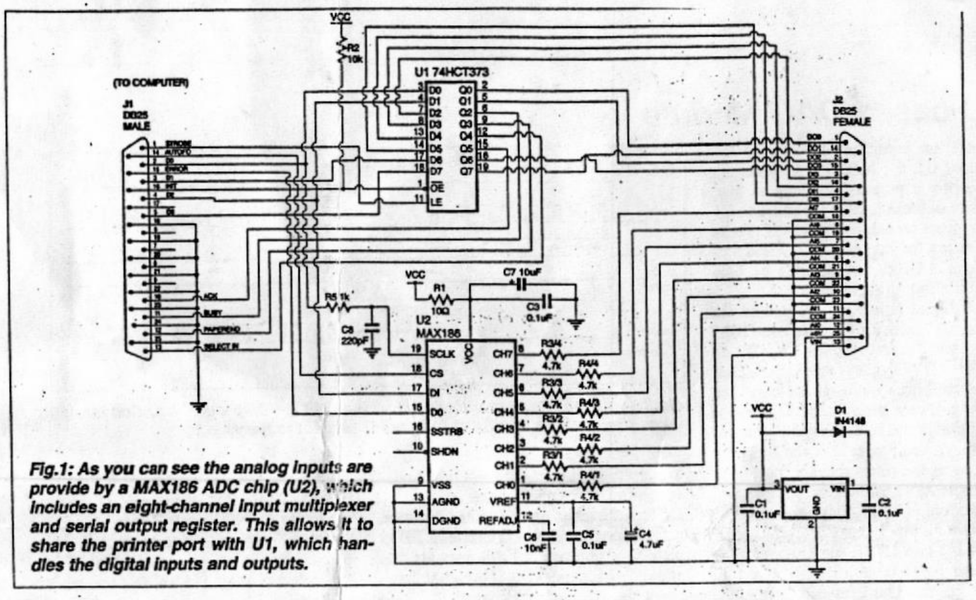


Fig.1: As you can see the analog inputs are provide by a MAX186 ADC chip (U2), which includes an eight-channel input multiplexer and serial output register. This allows it to share the printer port with U1, which handles the digital inputs and outputs.

Because I was using the parallel port, there was no way I could power the digital outputs from the parallel port. Hence it meant that the unit had to be externally powered. That in turn meant I had to fit in a voltage regulator — so a plug pack, battery or other type of supply could be used.

A PC's parallel port

The circuit diagram in Fig.1 shows the pins on the parallel port, at far left (J1). It is probably best if a short description of the parallel port is given. A more detailed description was given in *EA* for June 1995. A PC can normally have up to two parallel ports, 'LPT1' and 'LPT2', with the I/O base addresses shown in Table 1.

Each parallel port has three registers: a data register, a control register and a status register. The address of the data register is the base address, while the status register occupies the (Base address + 1) and the status register (Base address + 2). For LPT1 the addresses of the data, status and control registers would be 378H, 379H and 37AH respectively.

Each register is able to control or read the status of a number of the parallel port's lines. This information is listed in Table 2.

The parallel port's data register was originally designed to be 'write only', but in recent times most computers have been designed with the parallel port being bidirectional — hence it is both read and write. However to make this data acquisition unit operate with all types of PCs, the data register is used only as a write register. When you write a byte of information into the data register it appears on the lines D0 to D7 as shown in Table 2.

Table 1 — Base Addresses

Parallel Port	Address
LPT1	378H
LPT2	278H

The control register is again a write-only register. By writing a byte of information to this register you are controlling the printer 'handshake' lines Select, Init, AutoFeed and Strobe.

The final register is the status register, which is a read-only register. By reading this register you read the status of the handshake lines Busy, Ack, Paper End, Select In and Error.

Circuit description

Let's start off with the humble power supply. Power is fed into the unit via pins 13 and 24 of the female DB25 connector J2. Diode D1 is there to prevent damage if the user mistakenly reverses the connections. The voltage regulator is a 78L05, which converts the supply voltage to +5V. The +5V output is connected directly to the 74HCT373, but is filtered by resistor R1 and capacitor C1 before supplying the MAX186 ADC chip.

The +5V is also connected to pin 25 of the female DB25 connector. This allows the user to power external components such as pull-up resistors, when sensing the state of relays or switch contacts.

The 74HCT373 is enabled and disabled by the Init pin on the parallel port. To enable the device this line must be pulled low.

The data lines D0 to D3 on the paral-

Table 2 — Parallel Port Registers

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data	D7	D6	D5	D4	D3	D2	D1	D0
Status	Busy	Ack	Paper End	Select In	Error			
Control			Output Enable	IRQ Enable	Select	Init	AutoFd	Strobe

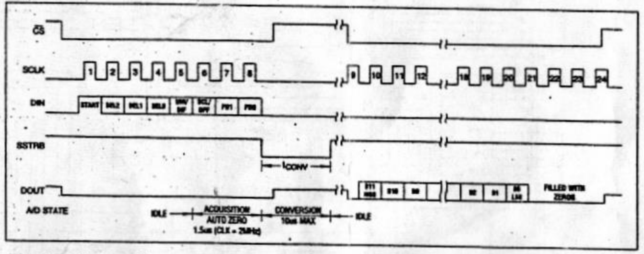


Fig.2: A timing diagram showing how the MAX186 chip operates. Conversion takes place after a control data byte is first sent to the chip.

lel port are connected to four of the inputs of U1, the 74HCT373. When the strobe line is pulled from high to low, the state of these lines is latched into U1 and appears on outputs Q0 to Q3. These outputs are fed into pins 1, 2, 3 and 4 of the female DB25 connector to become the unit's four digital outputs.

The digital inputs are brought in via pins 5, 6, 7 and 8 of the female DB25 connector. Taken to inputs D4 - D7 on the 74HCT373, they too are latched when the strobe pin goes low. The user's software then reads the state of the digital inputs by reading the status of the Busy, Ack, Paper End and Select In pins, using the status register.

U2, the ADC, is the most complicated of the chips. All communication is through the three pins CLK (clock), DI (Data In) and DO (Data Out). Data is clocked in and out using a series of clock pulses. The user first brings the CS (chip select) line low and then clocks in a byte of data on DI — which configures which channel on the ADC is to be read, whether the reading is to be single ended or differential, and whether the reading is to be unipolar or bipolar.

The user then waits approximately 10us for the ADC to finish its conversion, and then provides 12 clock pulses in order to read the ADC output on the DO line. A diagram of the process is shown in Fig.2. Table 3 shows the significance of the control byte fed to the MAX186 to configure its operation, while Table 4 shows how the SEL2, SEL1 and SEL0 bits are used to select the input channels in single-ended and differential modes.

The 4.7k resistors connected between each analog input and the female DB25 connector pins are to limit currents that could damage the ADC if a high voltage input is inadvertently applied. This allows the ADC to be able to withstand voltage transients up to +/-20V. However it should be noted that a voltage higher than 5V on any one input can affect the readings on the other inputs.

I had a problem with differential readings, in that at some areas within the ADC's range (e.g., around 1024 and 2048) it was showing an error of 20mV. Reading through the specification sheet I noticed Maxim states that when the ADC is carrying out a differential reading it uses a 'pseudo-differential' configuration, in that only the signal at the IN+ is sampled. The return side (IN-) must remain stable within +/-0.5 LSB (+/-0.1 LSB for best results) with respect to AGND during a conversion.

You're told to accomplish this by connecting a 0.1uF capacitor from AIN- (the selected analog input, respectively) to AGND.

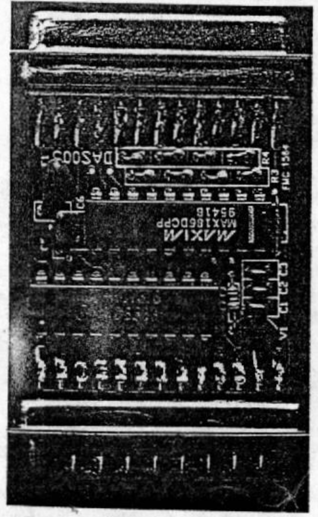
This is not a very practical situation for two reasons. Firstly, it is very hard to keep an input voltage that stable and in any case some voltage sources don't handle a 0.1uF capacitor on their output very well (they can oscillate and do other weird things). Secondly when measuring differential inputs, the magnitude of the positive input may be less than the negative input — hence you would expect the ADC to return a negative reading. In this situation the MAX186 just returns a '0'.

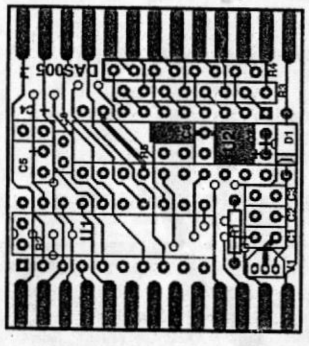
Because of this I have written the software so that when the user requests a differential reading, it actually does two consecutive single ended readings and then subtracts the readings. The problem with this method is that differential readings cannot be applied to high speed signals.

Construction

Remember throughout the construction that there is very little 'headroom' for the components when the DB gender changer hood is put into position. So make sure all of your components are small (the very small 1/4 watt resistors from Jaycar are perfect), and mount them as close to the board as possible. As you are fitting in the components it's a good idea to test that the hood will still fit in place before soldering.

The first step is to push the solder type 25-pin male and female D connectors on to the ends of the PCB board. Make sure you have the male connector at the end closest to the voltage regulator. Solder them into position. Note that in order to fit all of the components on to the board, a few of them have to be mounted on the bottom side. These components should be soldered into place first.





At left is a closeup of the prototype DAQ PCB, again just a little larger than actual size. The PCB overlay is shown above, for reference. Not surprisingly , there isn't a great deal of spare space — it's a fairly crowded little board, and care needs to be taken during assembly.

Turn the board over with the bottom side up and solder into position C7, C8, R2 and R5. Note the tantalum cap C7 will be bending over slightly, and C8 (220pF) should be small in size. When the legs are trimmed make sure they are cut as close as possible to the board.

Next turn the board with the top side up and solder in the rest of the capacitors and the resistors. R3 and R4 are SIL resistor arrays with four 4.7k resistors. Note that the tantalum capacitor C4 will again be bending over slightly. The final step at this stage is to mount the voltage regulator and the diode, making sure they orientated correctly before soldering.

Leave the 74HCT373 and the MAX186 chips out of the unit until we have tested the +5V supply.

Testing

To test the unit, first make up an input lead with a male DB25 connector fitted, with the connections as shown in Fig.2. Plug this onto the female input connector of the unit.

Do not connect the device to your PC port as yet. Instead carry out a final check to make sure all components such as tantalum capacitors, the diode and voltage regulator are correctly aligned.

Switch on the power to the unit and quickly check with a DVM that +5V is present between the pads for pins 20 and 10 on the 74HCT373. If this isn't the case there is something wrong with the regulator.

If you have got +5V, switch off the power and then solder in the 74HCT373 and the MAX186, making sure they are orientated correctly. I'm sorry there is no room for IC sockets with this unit. Now reapply the power and check the +5V again.

If everything is OK, switch off the power and connect the unit to the PC's parallel port. Switch on the PC and then the power to the data acquisition unit. Now run the DOS program LIBDAS5.EXE. This program reads all the analog inputs, digital inputs and outputs. You can now inject voltage signals to the analog inputs and read them, read the status of the digital inputs and watch the digital outputs being activated.

Table 3 - MAX186 Control Byte

Bit	Name	Description
7(MSB)	START	The first logic "1" bit after CS goes low defines the beginning of the control byte
6 5 4	SEL2 SEL1 SEL0	These three bits select which of the 8 channels is used for the conversion
3	UNI/BIP	1=Unipolar 0=Bipolar Can only be unipolar for this design as a bipolar power supply is required for Bipolar readings
2	SGL/DIF	1=Single Ended , 0=Differential
1 0(LSB)	PD1 PD0	Select clock and power down modes PD1=1 and PD=0 for Internal Clock mode.

The software

For those who wish to write their own programs, I have provided functions written in Borland C, QuickBASIC and Visual BASIC. Source code for these functions is provided. Sample programs (LIBDAS5.C and LIBDAS5.BAS) using the functions are included. Both these programs read all the analog and digital inputs and operate each of the digital outputs. Because of space restrictions in this article the source code and executables for these functions are not shown, but are available on *EA*'s Computer BBS — (02) 9353 0627.

For those people who want quick ease of use, we have produced a Windows program called I-SEE, which will run under Windows 3.1 or Windows 95. Written in Visual BASIC, this program allows the user to easily set up the data acquisition unit to view the analog and digital inputs, control the digital outputs as well as collect readings of temperature, pressure, the digital inputs and log them to disk.

In summary, I-SEE allows the user to:

- View the status of the digital inputs;
- Control the status of the digital outputs;
- View the analog inputs. The readings can be viewed and logged as 'raw' readings of 0 - 4096, or they can be scaled to engineering units: e.g., 0 - 100°C or 0 - 1000kPa;
- Display the analog inputs on graphs or trends. The user has access to four graphs and on each of those graphs the user can plot any four analog inputs;
- Having an inherent accuracy of +/- 3mV, the unit can easily be calibrated for more accurate tasks; and
- Log or store the analog input and digital input readings to a file on your hard disk or a floppy disk.

Table 4- MAX186 Channel Selection

SEL 2	SEL 1	SEL 0	SE Channel	Diff Channel
0	0	0	0	0+ 1-
1	0	0	1	1- 0+
0	0	1	2	2+ 3-
1	0	1	3	2- 3+
0	1	0	4	4+ 5-
1	1	0	5	5+ 4-

The logging of data to disk has the following features:

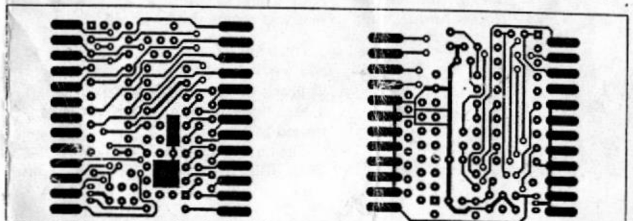
1. The period between readings can be varied between 1 and 32767 seconds;
2. The data is stored in comma-delimited format, allowing the user to load the data into an Excel or Lotus 123 spreadsheet easily — where it can be manipulated, graphed and reports made;
3. Storage of data can be initiated and stopped by the user clicking a box on the screen, or the user can program data storage to commence and end when a nominated digital or analog input reaches a set value.

The I-SEE program is supplied by Ocean Controls with the DAQ module or PCB — see the next section. Note that the source code for I-SEE is not included; only an executable.

Module or PCB available

The Data Acquisition and Control unit is available as a complete module, including all software, from Ocean Controls, of 4 Ferguson Drive, Balnarring 3926; phone (059) 83 1163. The complete module with software is priced at $120. Alternatively just the PCB and software are available, for $45.

Ocean Controls also has a DB25 connector to screw terminal interface board, available to allow easy connection to the Data Acquisition Unit. This can either be mounted in a jiffy box or on spacers. ◆



The etching patterns for both sides of the DAQ module PCB, reproduced here actual size for those who like to etch their own.