

---

# A low-cost interface circuit to enable A/D conversion using the parallel port

M. Rodríguez<sup>1</sup>, A. Ayala<sup>1</sup>, F. Rosa<sup>1</sup>, F. Herrera<sup>1</sup>, S. Rodríguez<sup>1</sup>, M. Díaz-González<sup>2</sup> and S. R. Buenafuente<sup>1</sup>

<sup>1</sup>*Department of Fundamental and Experimental Physics, University of La Laguna, Tenerife, Canary Islands, Spain*

<sup>2</sup>*Department of Animal Biology, University of La Laguna, Tenerife, Canary Islands, Spain*

*E-mail: mrvalido@ull.es*

**Abstract** A simple and low-cost acquisition device using the computer parallel port has been used to implement a full acquisition system composed of an analogue interface, a 12-bit analogue-to-digital converter, a quad three to one multiplexer and a software package aimed at managing the acquisition process. The features of this system make its implementation a suitable educational resource for undergraduate students from Engineering, Physics and Computer Sciences faculties.

**Keywords** A/D conversion; interface; signal acquisition; Standard Parallel Port

The parallel port is an interface commonly used to connect inexpensive and powerful devices which allows the development of a number of diverse tasks, such as LCD control, PC to PC communication, Ethernet link, data logging, motor control, etc. This port provides eight TTL outputs, five inputs and four bi-directional leads that allow the input of up to 9 bits and the output of 12 bits at a time, requiring minimal external circuitry to implement desired tasks.

In 1994, the Institute of Electrical and Electronics Engineers (IEEE) established a new standard to manage the bi-directional signalling methods of the parallel interface of personal computers.<sup>1</sup> This standard provided a high speed bi-directional communication between the PC and any external peripheral. It could communicate 50 to 100 times faster than using the original parallel port, but remained fully compatible with this previous version. This standard, referred to as standard 1284, defined five operational modes to enable communication with external peripheral or data transfer. Each mode fixed a procedure for transferring data in either the forward direction (PC to peripheral), the reverse direction (peripheral to PC) or bi-directional (half duplex). These defined modes are named *Compatibility*, *Nibble*, *Byte*, *EPP* (Enhanced Parallel Port) and *ECP* (Extended Capabilities Port) modes.

The *Compatibility* mode, also called Standard Parallel Port (*SPP*) or 'Centronic Mode', is a forward direction only mode that is used to drive the printer devices. For reverse direction only, there exist two modes, the *Nibble* mode that uses 4-bit status lines for data at a time, and the *Byte* mode that uses 8-bit data lines for data transfer. The latter is sometimes referred to as the IBM PS/2 bi-directional port. Finally, among the Bi-directional modes, *EPP* is mainly used for driving external CD-ROM, tapes, hard drives, network adapters, etc., while *ECP* is primarily used by new generation printers and scanners.

The *Nibble* mode is the most common way to get reverse channel data from a printer or peripheral device. This mode is usually combined with the *Compatibility* mode to create a complete bi-directional channel. All standard parallel ports provide five lines from the peripheral to the PC that are used for external status indications. Using four of these lines, an external device can send data by means of the information nibbles (4-bits) to the PC.

Due to the fact that all personal computers can be configured in *Compatibility* mode, we have developed a data acquisition system that can be widely installed in all PCs. The system is basically composed of an input stage, which acts as a signal conditioner that allows the adjustment of the gain and offset of the analogue signal input. This conditioned signal is set up to 12-bits A/D converter, where it is digitised. Since the port configured in *SPP* mode has only five input lines (of which we have used four), we have placed a multiplexer between the A/D converter and these lines to split the 12-bit word into 4 input bits. In this way, a word is read in three consecutive steps. In order to manage the acquisition system we have developed a software package, which, in addition to controlling the transfer process, also displays and stores the data. The design and implementation of the system has been used to teach to the students the basic concepts about analogue signal conditioners, A/D conversion and how an interface device with the parallel port works so as to implement the software necessary to acquire data through this port. The system was satisfactorily tested by acquiring well-known generated signals such as square, triangle and sine waves.

### Parallel port basic description

Commonly, the PC printer port has four control lines, five status lines and eight data lines. It is usually found on the rear part of the PC as a D-Type 25-pin female connector. Table 1 shows the correspondence between connector pins and signal names; they are also described in terms of their input/output

TABLE 1 Pin out and signal names of the parallel port

Pin number	SPP signal	PC In/Out	Register	Hardware inverted
1	$\overline{\text{Strobe}}$	In/Out	Control	Yes
2–9	$\text{Data}[0..7]$	Out	Data	No
10	$\overline{\text{Ack}}$	In	Status	No
11	$\overline{\text{Busy}}$	In	Status	Yes
12	$\overline{\text{PaperEmpty}}$	In	Status	No
13	$\overline{\text{Select}}$	In	Status	No
14	$\overline{\text{AutoFeed}}$	In/Out	Control	Yes
15	$\overline{\text{Error}}$	In	Status	No
16	$\overline{\text{Init}}$	In/Out	Control	No
17	$\overline{\text{Select-Input}}$	In/Out	Control	Yes
18–25	Gnd	—	—	—

condition, from the point of view of the PC, and register to whom signals belong. The *SPP* has 3 base addresses that are associated with LPT1, LPT2 and LPT3 device labels respectively. Normally when the computer is turned on, the Basic Input Output System (BIOS) assigns the base addresses 378H-37FH to LPT1 and, in the case that additional parallel ports exist, the BIOS allocates the addresses 278H-27FH to LPT2 and so on.<sup>1,2</sup>

The functionality of a parallel port is achieved through the use of 3 addressable ports which are shown in Table 2, together with their associated registers and control gatings. The base address, usually called *data port* or *data register*, is simply used for outputting data on the parallel port data lines. The *status port* (base address + 1) is made up of five input lines (*Busy*, *Ack*, *Paper Out*, *Select In* and *Error*), an IRQ signal and two reserved bits. All these lines notify the CPU of the printer status and can only be read by it. The *control port* (base address + 2) is a read/write register and endowed with four functions, i.e., Strobe, Auto line feed, Initialise and Select printer.

### Interface circuit

In order to acquire and process any analogue signal into a PC it is necessary to have an interface that allows the conversion of these signals into digital form. The interface board implemented in this work is a simple circuit which is composed by a signal conditioner, an A/D converter and a digital adapter that allows the connection of the converter output to the parallel port of the personal computer (Fig. 1).

TABLE 2 Map port addresses for the Standard Parallel Port (SPP)

Port name	Direction
Data Port	Base Address + 00H
Status Port	Base Address + 01H
Control Port	Base Address + 02H

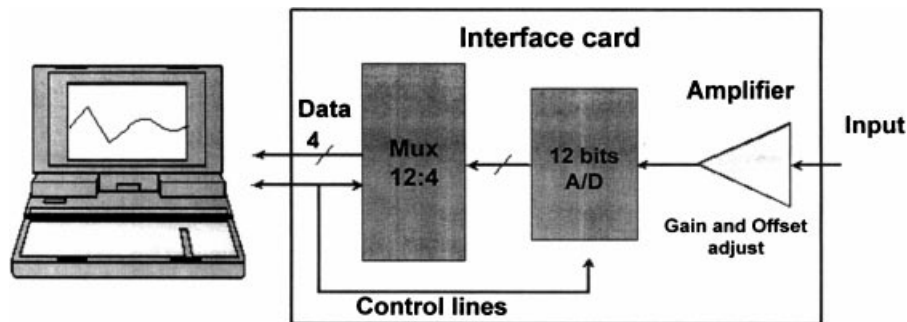


Fig. 1 Block diagram of the system.

A schematic view of the electronic circuit is depicted in Fig. 2. As can be seen, it comprises three integrated circuits and few discrete components.

The input stage contains a dual operational amplifier (TL082) which was used to implement a buffer necessary to insulate the voltage divider output (U3B) of the differential amplifier inverted input (U3A).<sup>3</sup> This circuit allows the adjustment of the amplitude and offset values of the analogue signal to the input voltage windows ( $\pm 10$  V) of the A/D converter in order to obtain the best A/D conversion signal-to-noise ratio.

The conditioned signal is digitised by the A/D converter, which is configured in bipolar mode with an accuracy of 12 bits (ADS7804).<sup>4-6</sup> Furthermore, as we mentioned before, the *nibble* mode of the port reads data through the port without setting it in the reverse mode. In this mode, it is necessary to use a switch circuit that splits the word into three nibbles (4 bits) to acquire a 12-bit word from the A/D converter. With this aim, we have implemented a quad three line to one multiplexer built in the Programmable Logic Device (PAL16L8) to connect the A/D output to four out of five status lines, which are selected as data input lines. Table 3 shows the Standard Parallel Port signal labels together with the corresponding names and functions in our system. For example, *Strobe* is labelled as *Read/Convert* and its function is to set up the A/D converter in either read or convert mode, this signal is handled by the PC timer which controls the sample frequency through a call to a low-level routine. Hence, when the A/D converter is set in read mode, the 12-bit word located in its output register are multiplexed in three 4-bit words. This process is controlled by the control lines *S0* and *S1*, and the output of the multiplexer is connected to *Ack*, *PaperEmpty*, *Select* and *Error*.

Figure 3 shows the data transfer cycle between the external peripheral and the PC. The conversion starts when the PC asserts the converter pulse (signal *Read/Convert* low) during a time longer than 40 ns. Then the external device sets the *Busy* signal to low during the conversion time (maximal 8  $\mu$ s) and thereafter sets it to high. The *Busy* signal high indicates to the PC that the data are valid and therefore the three nibbles are consecutively selected to be read, asserting the multiplexer control lines, *S0* and *S1* to values 00, 01, 11. This cycle is repeated whenever a new interrupt occurs.

### Control software

Fully dedicated software was developed to control the data acquisition system. The software was written in Turbo Assembler and C (Borland C, version 3.0) to run under DOS 5.0 or higher on an IBM compatible PC or laptop computer with minimal hardware requirements (80286-microprocessor, 16 MHz, 1 Mb RAM and VGA graphic card). The program is made up of a main function and several modules managing the low-level routines (timer programming, parallel port configuration and hard disk storage), the graphic user interface (GUI) and the statistical data analysis. The flow chart corresponding to the main function is illustrated in Fig. 4. Once the software is launched, a subset

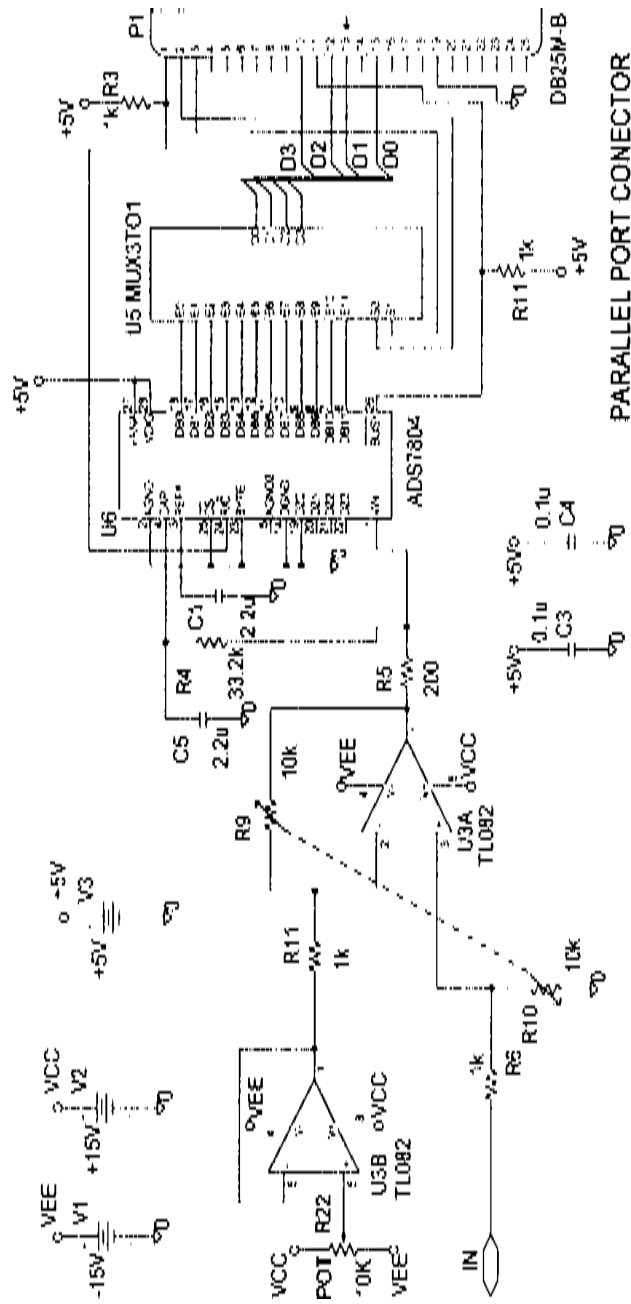
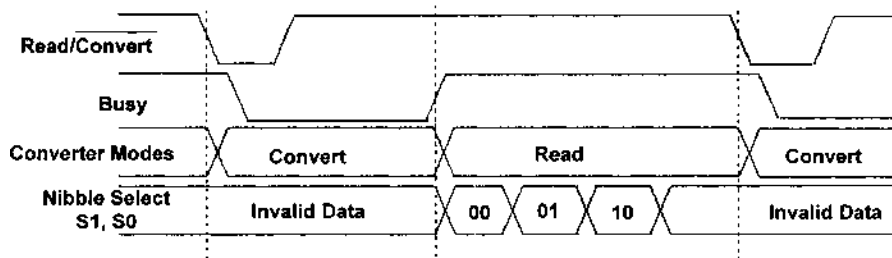


Fig. 2 Schematic view of the interface circuit.

TABLE 3 *Functionality and equivalence between SPP signals and nibble mode*

SPP signal	Nibble mode name	PC In/Out	Description
$\overline{\text{Strobe}}$	$\text{Read}/\overline{\text{Convert}}$	Out	Set the converter to Acquire/Converter mode
$\overline{\text{Busy}}$	Busy	In	When set up to low (from converter) the data are available
$\overline{\text{Ack}}$	D3	In	Used for data bit 3, 7 and 11
$\overline{\text{PaperEmpty}}$	D2	In	Used for data bit 2, 6 and 10
$\overline{\text{Select}}$	D1	In	Used for data bit 1, 5 and 9
$\overline{\text{Error}}$	D0	In	Used for data bit 0, 4 and 8
$\overline{\text{Data}}[0:1]$	$S[0:1]$	In	Select the nibble to be read
$\overline{\text{AutoFeed}}$	—	Out	Not used
$\overline{\text{Select-Input}}$	—	Out	Not used
$\overline{\text{Init}}$	—	Out	Not used
$\overline{\text{Data}}[2:7]$	—	In	Not used

Fig. 3 *Data transfer cycle during one conversion and read process.*

of menus allows the user to introduce the required data to begin the acquisition process, i.e. the sample frequency, the acquisition time and the name of the file where data will be stored. The GUI was created so that besides the menu options, the real-time signal plot, the acquisition and the signal parameters are also shown in Fig. 5.

The acquisition process starts with the PC timer programming, which allows the setting up of the sampling frequency. After that, the parallel port is configured in simple mode, so that the 12-bit word is read in three steps, as explained above. Once the acquisition parameters have been configured, a background routine starts the signal acquisition process. The acquisition parameters may also be loaded from a default configuration file, which is particularly useful once the acquisition values have been optimised, allowing the user to avoid the initial configuration steps.

Other routines are launched once the acquisition process has started. The first accomplishes the real-time signal screen display, and the second routine computes the main statistical parameters of the signal, such as mean, standard deviation, maximum and minimum, as well as the period. The statistical parameters and the voltage range of the signal allow the user to make fine adjustments in the gain in the amplification stage. Moreover, simple keystrokes allow

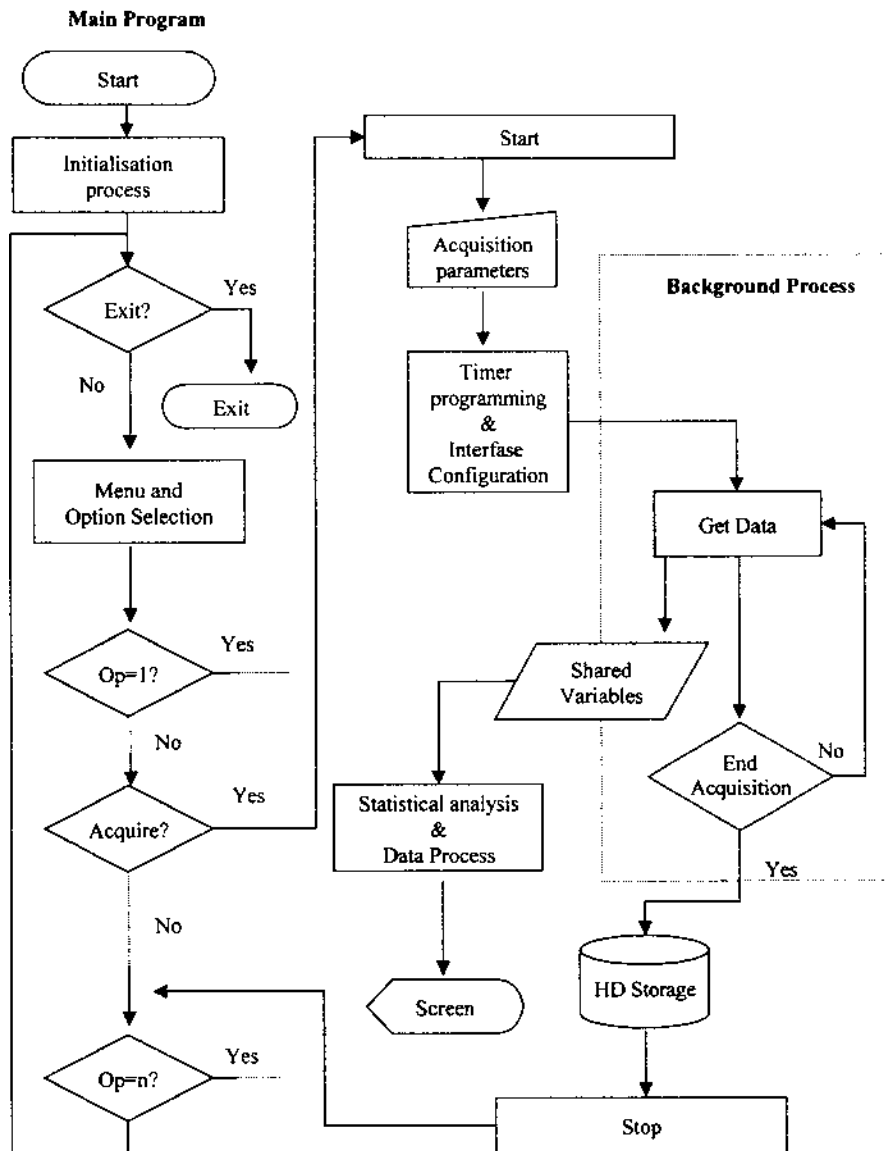


Fig. 4 Simplified flow chart of the control software.

changes to be made to the display options and setting of the voltage and time scales of the graphic output.

Another module undertakes the hard disk data storage. The acquired data are stored in a reserved buffer of the RAM memory and passed to a fixed memory once the acquisition is completed. The software was programmed so that the data storage routines make an estimation of all the RAM memory

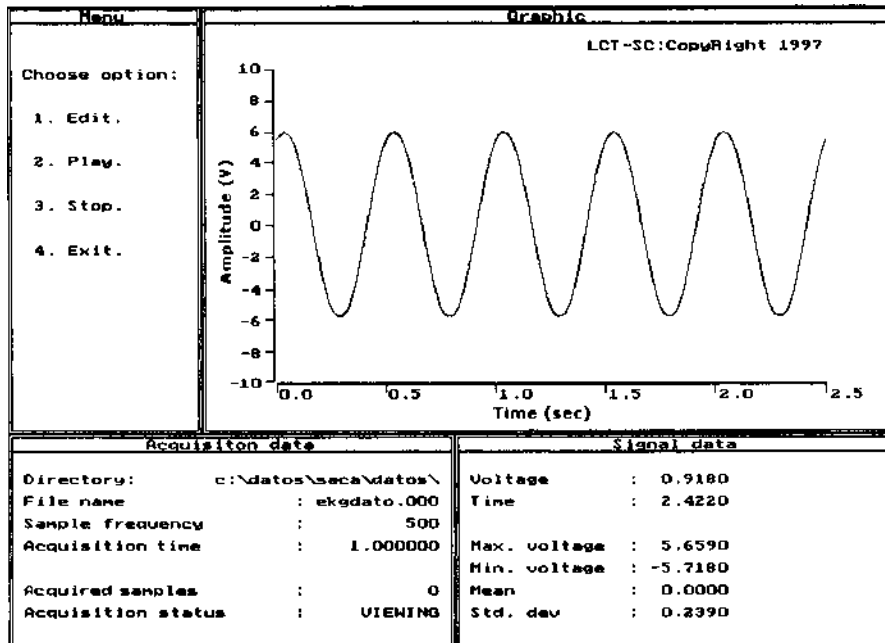


Fig. 5 A screen shot of the program.

available (including extended memory) and accordingly, the buffer is resized. Obviously, the recording time is conditioned by the acquisition frequency. Finally, from the main menu, a subset of submenus includes options to replay stored files and to export or print the signal data (Fig. 5).

### Educational usefulness

Given that the whole system described above involves hardware and software design aimed at devising a full acquisition system, it may ideally be conceived as a problem-based laboratory exercise suitable for a team project of about 4–6 students during one semester. Every student group would have to compile the specifications of the project, understand them and make up a developing plan. This first stage should be done under the supervision of the tutors. In the next step, different tasks would be assigned to the students making up a team, i.e. devise the hardware (interface, A/D converter and multiplexer) and write the control software allowing operation of the interface. Finally, testing, debugging and the integration would allow full implementation of the acquisition system. All steps are suitable for evaluation, providing an excellent educational resource that offers an opportunity to reinforce primary teaching methods, such as lecturing and seminars.

## Conclusion

A full portable, simple and low-cost acquisition system based on the communication features of the PC parallel port has been implemented. The interface circuit, which was designed with few electronic components, makes it an affordable option to accomplish the conversion of analogue signals into digital inputs to PCs. The specific acquisition software developed here provides a useful tool to manage the acquisition process. Using several kinds of signals, like sine, triangle and square waves as paradigms of analogue signals, the accuracy and performance of all system modules were successfully tested.

Finally, we have explored the adequacy of the acquisition system for educational purposes by assessing the attitudes of academic staff and undergraduate students from the Faculties of Physics, Engineering and Computer Sciences at the University of La Laguna, towards the potential learning value of the prototype. Although the survey was not evaluated in detail, it was generally considered that the system could be used as a complementary practical tool for illustrating complex concepts concerning parallel port programming, A/D conversion and dedicated circuit development.

## Acknowledgements

We are very grateful to Sharon Hardy for her helpful comments and critical reviewing of the manuscript. This work was partially funded by a grant from Gobierno Autónomo de Canarias (Spain).

## References

- 1 IEEE, *IEEE 1284–1994 Standard*.
- 2 Michael Tischers, *PC Interno. Programación del Sistema* (Marcombo Boixareu Editores, Barcelona, 1995), pp. 276–310.
- 3 P. Horowitz and W. Hill, *The Art of Electronics* (Cambridge University Press, New York, 1989), pp. 178–185.
- 4 Burr-Brown, *IC Data Book, Data Conversion Products* (Burr-Brown, Tucson, 1995).
- 5 B. M. Gordon, *The Analogic Data-Conversion System Digest* (Analogic Corporation, 1977), pp. 6–21.
- 6 Analogic Device, *Analogic-Digital Conversion Notes* (Analogic Device, 1977), pp. 7–22.