Instruments and Experimental Techniques, Vol. 48, No. 5, 2005, pp. 608–610. Translated from Pribory i Tekhnika Eksperimenta, No. 5, 2005, pp. 53–55.

APPLICATION OF COMPUTERS IN EXPERIMENTS

Interfacing the Instrumental GPIB with a Personal Computer Through the LPT Port

Yu. A. Semerenko

Verkin Institute for Low Temperature Physics and Engineering, National Academy of Sciences of Ukraine, pr. Lenina 47, Kharkov, 61103 Ukraine
E-mail: semerenko@ilt.kharkov.ua

Received March 15, 2005

Abstract–A device for servicing a GPIB is described. The display registers of the LPT port are used, and, therefore, the GPIB interface is also capable of operating on old SPP-version cards.

As a rule, up-to-date programmable controlling and measuring devices are equipped with a so-called general-purpose interface bus (GPIB; its domestic analog is КОП [1]), which is used as the main data transmission channel. Devices satisfying the IEEE-488.2 hardware and software standard specifications can be connected to a personal computer (PC) using standard interface cards and drivers of the popular instrumental LabView or LabWindows platforms. Unfortunately, most domestic devices have been developed in accordance with the IEEE-488.1 hardware standard (compatibility of electrical and mechanical parameters, protocol of low-level interaction with the bus, and absence of a standardized system of commands); hence, there are no standard drivers for them. It can prove inexpedient to connect these devices to the PC by using standard interface cards, since they need special, complicated software.

The soft hardware emulation of GPIB interface functions by using the adapter connected to the parallel LPT port of a PC [2, 3] can become an alternative method for building a computer-aided measurement system. The earlier described devices of this type have some drawbacks. Thus, three parallel ports are required to control the device [2], while the standard design of most modern PCs includes only one. The device [3] does not completely satisfy the IEEE-488.1 standard electrical compatibility specifications; as a result, it has insufficient noise immunity and cannot operate as part of a system with several devices. In addition, the impossibility of controlling the *IFC* and *REN* lines does not allow one to use this device in systems with several controllers.

The device described below is free from the above drawbacks. Only nine lines of the LPT port are necessary to control it. This fact allows one to employ the remaining lines for servicing other devices that do not have the GPIB, for example, by using the device [4]. The described device (see figure) consists of the GPIB-matching assembly based on microcircuits M_9–M_{14}, galvanic decoupling assembly with the PC parallel port (microcircuits M_{15}, $M_{11\text{-}5}$, and $M_{11\text{-}6}$, and *Op1–Op8*), and bidirectional parallel-to-serial converter with buffer RAM (microcircuits M_1–M_8).

In order to set the necessary combination of logical levels on the GPIB lines, it is necessary that the corresponding sequence of logic zeros and units be consecutively written into registers M_3 and M_4 and, after that, that the software form a short positive pulse on line *D7* of the LPT port, in response to the leading edge of which the data written into registers M_3 and M_4 will be rewritten into the buffer RAM (M_7 and M_8) and presented on the corresponding lines of the GPIB.

The algorithm of the consecutive writing into registers M_3 and M_4 consists of the following sequence of operations: (1) a short negative pulse clearing registers M_3 and M_4 from the previous information is formed on line *D5* of the LPT port, and (2) the required combination of 16 logic zeros and units is consecutively set on line *D3* of the LPT port, and, in this case, when each next signal is set, a short positive pulse is formed on the line *D6* of the LPT port, in response to the leading edge of which the logic level set at input 1 of M_3 is consecutively bit-by-bit written into registers M_3 and M_4. It is necessary to take into account that the device inverts the signal applied at its input, i.e., if a high TTL level is applied to line *D3* of the LPT port, a low TTL level will be set on the appropriate GPIB line, which, in accordance with the inverse TTL logic used in the GPIB standard, corresponds to a logic unit. To ensure correct operation of the device, the control program should stipulate that the logic zeros (high TTL level is on the relevant lines of the bus GPIB) be written into all registers M_3 and M_4 corresponding to inactive (at that moment) GPIB lines.

In order to read the current state of the GPIB, it is necessary to execute the following sequence of actions: (1) a short positive pulse is formed on the line *D2* of the LPT port, in the response to the leading edge of which the current state of the GPIB lines will be fixed into the

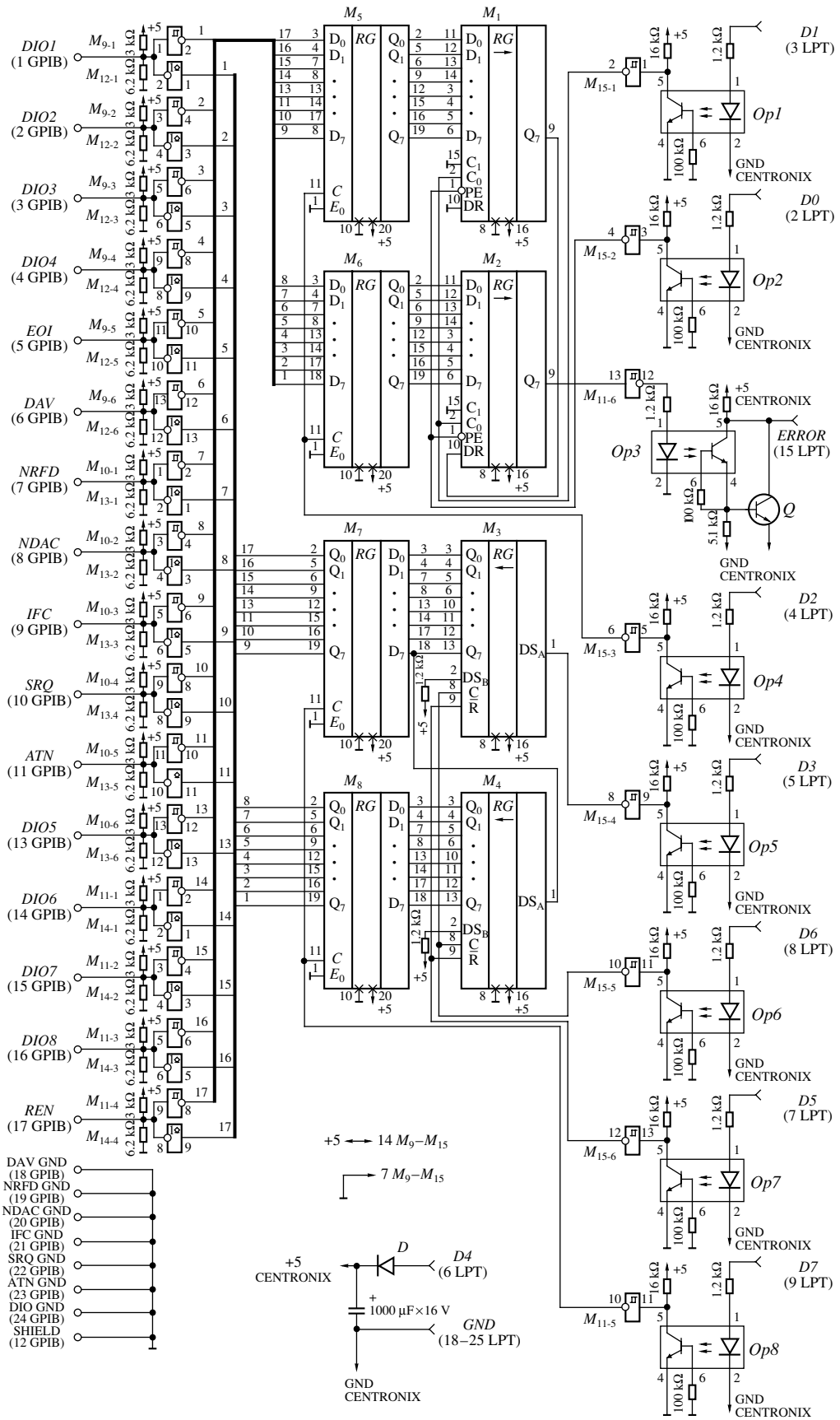

Fig. 1. Schematic diagram of the servicing GPIB bus: (M_1, M_2) IN74HC165AN, (M_3, M_4) IN74HC164AN, (M_5–M_8) IN74HC374AN, (M_9–M_{11}, M_{15}) IN74HC14AN, (M_{12}–M_{14}) IN74HC05AN, (*Op1*–*Op8*) 4N35, (*Q*) S9014, and (*D*) 1N4148.

buffer RAM (M_5 and M_6); (2) a short negative pulse is formed on the line *D0* of the LPT port, in response to the leading edge of which the contents of the buffer RAM (M_5 and M_6) will be written into the series–parallel registers M_1 and M_2; (3) and when a high TTL level is set on the line *D0* of the LPT port, a sequence of 15 positive pulses should be formed on line *D1* of the LPT port. In response to their leading edges, the data stored in registers M_1 and M_2 are consecutively bit-by-bit transmitted to pin 9 of register M_2; in this case, the logic levels responding to the data written into M_1 and M_2 are consecutively set on line *ERROR* of the LPT port. The order of the data transmission is as follows: the last register D_7 of microcircuit $M_2 \longrightarrow$ register D_1 of microcircuit M_1. Thus, in order to read data from the adapter, before the transmission of each subsequent pulse to *D0*, the control program should read the state of the register of the LPT port servicing the line *ERROR*.

The duration of all control pulses should be ≥40 ns.

Formally, the GPIB standard allows one to integrate up to 15 addressed devices into a computer-aided measurement system. However, in fact, 31 unique addresses are admissible. Hence, when operating devices that do not use the IEEE-488.2 protocol, their total number in the system can be increased up to 31.

While developing a control program, special attention should be paid to the correct realization of the HANDSHAKING data transmission algorithm and to the processing of the interface control signal *SRQ*. In addition, one should remember that only one device can be in the data transmission state at any particular moment and that an arbitrary number of devices can be addressed to the data receipt. The purpose of the signal lines of the GPIB, protocol of interaction with the bus, method for addressing the switched-on devices, and basic data coding principles are described in detail in [5].

The arbitrary number of controllers can be simultaneously connected to the GPIB (only one of them will be active). This circumstance allowed me to use one B2-38 direct current nanovoltmeter (with the corresponding switching of the input circuits) as part of two simultaneously operating experimental setups, each having its control computer.

Several samples of the described device have been manufactured, and they operate with INTEL-486-based PCs as part of experimental setups intended for studying the resistive, magnetic, and dielectric characteristics of materials at low temperatures. The software is written in Turbo Pascal 7.0.

The device consumes a current of ≤0.6 A and can be powered from the power supply of the PC.

REFERENCES

1. GOST (State Standard) *26.003-80 (ST SEV 2740-80): ESSP Interface System for Measuring Devices with Byte-Sequenced and Bit-Parallel Exchange of Information. Requirements for Compatibility*, 1980.
2. Knysh, A. and Teslenko, A., *Radio*, 1995, no. 12, p. 26.
3. Nadtochii, A.B., *Prib. Tekh. Eksp.*, 2003, no. 6, p. 138.
4. Semerenko, Yu.A., *Prib. Tekh. Eksp.*, 2005, no. 3, p. 162.
5. *Interfeis dlya programmiruemykh priborov v sistemakh avtomatizatsii eksperimenta* (Interface for Programmable Instruments in Experiment Automation Systems), Moscow: Nauka, 1981.