TECHNICAL NOTE

AN INTELLIGENT COMPUTER INTERFACE UTILIZING PARALLEL PICOCONTROLLERS

Michael Collier

Department of Electronic Engineering, National University of Science and Technology Bulawayo, Zimbabwe, collier1942@yahoo.co.uk

(Received: November 29, 2001 – Accepted in Final Form: September 10, 2002)

Abstract The design of an interface unit is described, in which RS232 serial data is converted to latched parallel data on 22 independent lines. The data direction of each line is programmable through the serial port. Two picocontrollers are employed in a parallel processing mode to give the required number of I/O pins, and data on the shared serial line is coded to separate data streams to the individual picocontrollers, and to avoid contention. The hardware configuration and the software for both the computer and microcontroller are described. The speed penalty of such a system compared with direct bus connection is analyzed.

Key Words Computer Interface, Picocontroller, Serial Communication

چکیده طرح یک مدار واسطه توصیف شده است که در آن داده های سریال RS232 به داده های موازی Latch شده روی ۲۲ خط مستقل از هم تبدیل شده است. جهت داده های هر خط از طریق دهنه (Port) سریال قابل برنامه ریزی است. دو پیکوکنترولر در مَد پردازش موازی بکار گرفنه شده اند تا تعداد پین های I/O مورد نیاز را بدست دهند و داده های روی خط سریال مشترک (Shared) به منظور جداسازی جریان داده ها به هر یک از پیکو کنترولرها کد بندی شده است تا از در هم ریختگی پرهیز گردد. تشکیلات سخت افزاری و نرم افزاری هم برای رایانه و هم برای میکروکنترولر توصیف شده است. جریمه سرعت چنین سیستمی در مقایسه با اتصال مستقیم باس (Bus) تحلیل گردیده است.

1. INTRODUCTION

Data-collection systems in the industrial, agricultural, meteorological and medical spheres often require the communication of binary data to a personal computer through an appropriate interface. The classical solution is the addition of a plug-in card in the computer, which provides a parallel bi-directional interface. However, there are some situations in which portability, price and flexibility make the use of the serial port attractive, if external circuitry achieves the conversion to parallel format.

In a particular industrial application in Zimbabwe involving automatic testing [1], an interface to a personal computer was required for 22 bits of bi-directional parallel data. To meet some of the above requirements, the interface has been implemented by use of microcontroller-based external circuitry. To provide the specified number of data lines, two picocontrollers are operated in parallel, but configured to share the same serial port. The use of coded serial bytes allows data to and from the individual picocontrollers to be recognized and handled appropriately.

2. THE HARDWARE CONFIGURATION

As shown in Figure 1, two picocontrollers share the same Transmit and Receive lines to the host computer. The differentiation of messages for the individual PIC is achieved by the logical circuit

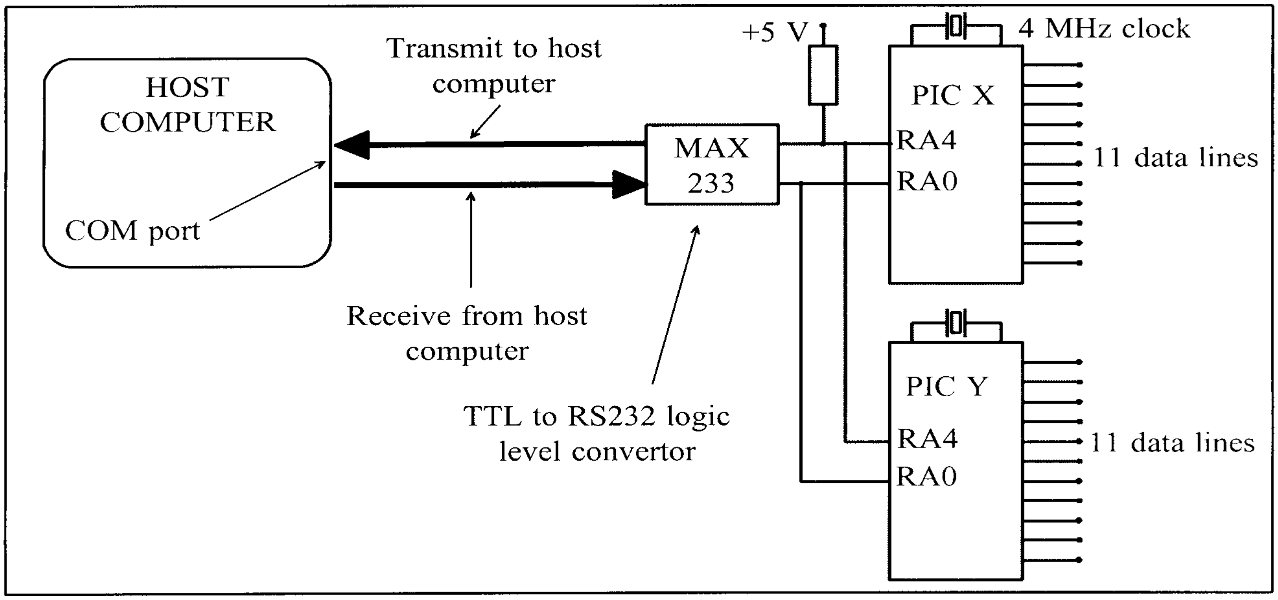

Figure 1. The hardware circuitry.

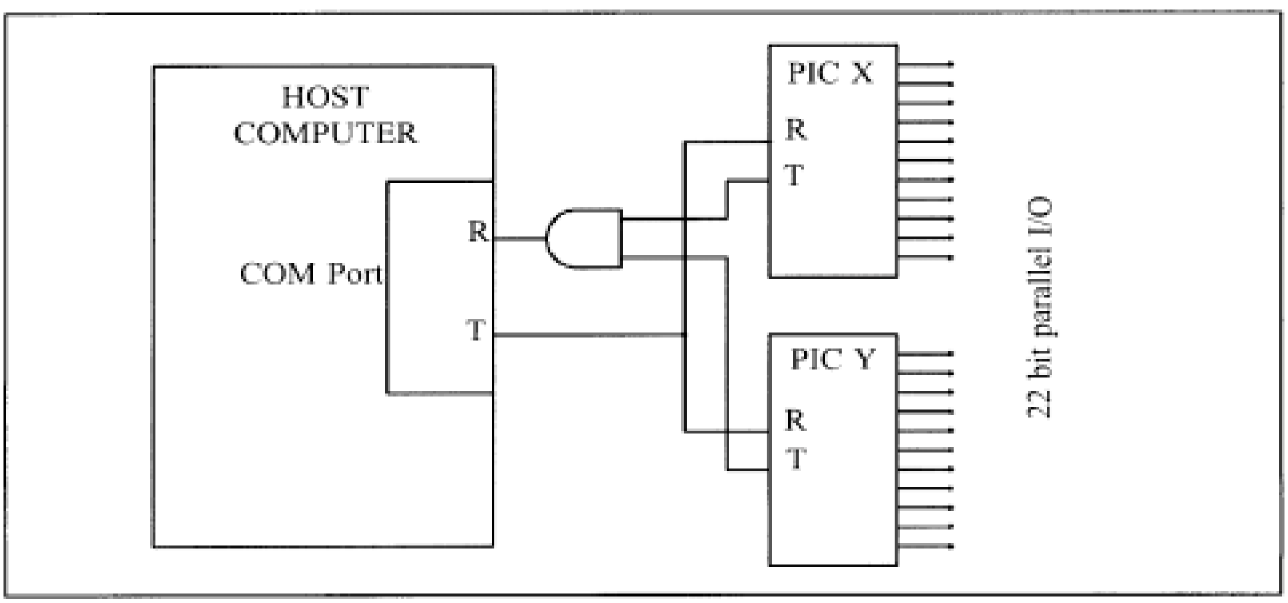

Figure 2. Logical equivalent circuit.

connection and by coding of bytes from the host computer.

The version of picocontroller selected for this application is the PIC 16C84 from Microchip, in which the program is stored in EEPROM [2]. The I/O facilities comprise two ports, where Port A has five lines (RAO-RA4), and Port B has eight lines (RBO-RB7). As shown above the two PICs are

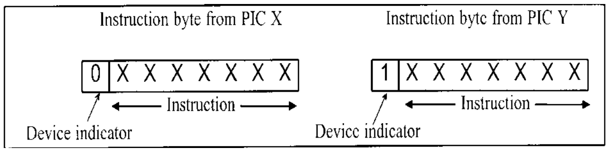

Figure 3. Instruction Bytes transmitted by the computer.

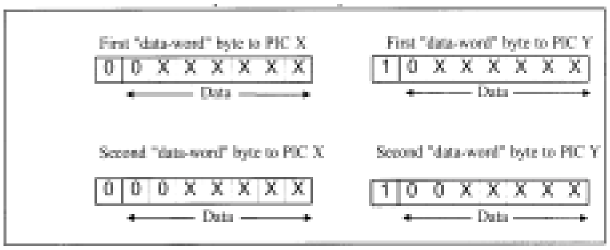

Figure 4. "Data-word" bytes transmitted by the computer.

connected such that the serial connections use the RAO lines for receiving and the RA4 lines for transmitting. The RA4 line is chosen because it has an open-collector connection, which permits the setting up of a logical AND between signals from the two PICs, when a common load resistor is provided for the two transmit lines. The serial connections then become logically equivalent to the circuit of Figure 2. In this situation an "idle" condition on one PIC, which is logic 1 for the RS232 protocol, will allow the other PIC to transmit.

The remaining eleven data lines on each PIC are then available as bi-directional latched I/O lines. To convert the serial signals to proper RS232 levels a MAX233 chip is used, as shown.

3. DATA CODING

3.1 The Master-Slave Concept Each transfer of data is initiated by an instruction byte from the computer, which is then acknowledged by the appropriate PIC Y. Thus, contention on the serial port is avoided, since the possibility of the PICs transmitting simultaneously is eliminated, because neither PIC will initiate a transfer of data unless polled by the computer.

Each byte from the computer is coded to indicate the intended recipient PIC, and the MSB is used as the indicator bit, where a zero signifies PlC X, and a one PIC Y. The remaining seven bits of each instruction byte are then available for passing 128 possible different instructions to each of the PICs. Upon receipt of an instruction byte, the relevant PIC replies with a pair of bytes comprising either data or an error message. Thus, handshaking for each instruction sent by the computer is achieved. The form of the instruction bytes is shown in Figure 3.

3.2 Transmitting Data from the Computer to the Parallel Port When the computer is sending data, the instruction byte is followed by four "data words". Each of these comprises a device-indicator bit (MSB) followed by five or six bits of data. Thus four of these "data-words", two to each PlC, allow 22 bits of data to be communicated to the interface, which latches the appropriate pins with the logic levels. The form of these "data-word" bytes is shown in Figure 4.

3.3 Data Reading by the Computer from the PICs To initiate a read operation, the computer sends an appropriate instruction byte to first one PlC and then the other. Upon receipt of a byte, the appropriate PlC then sends back a pair of bytes. These contain data, in the format shown in Figure 4, representing the logic levels on the eleven pins of the PlC parallel port. It will be noticed that "data-word" bytes always have a zero in bit 6 (the bit below the MSB). If a PIC puts a one in that position, it indicates that the byte contains an error-message, which the computer will interpret and use accordingly.

4. SOFTWARE DESIGN

4.1 The Driver Program for the PC The driver software in the host computer comes in the form of a C function "pic_instruction (char instr)", which takes a byte as the argument, and returns a long integer when reading data into the computer. This function can be included into application programs requiring communication with the interface, using character arguments to indicate the specific instructions to be executed. From this, other functions are called which provide the coding of data, access to the COM port, and communication with the PICs.

4.2 The Resident PlC Program The PlC source program is written in Parallax assembly language, containing routines to examine the byte streams being received. After identifying an instruction intended for it, a PIC will then handle the reading or writing of data between its eleven parallel pins and the computer port.

5. DATA TRANSMISSION SPEED IMPLICATIONS

Obviously this arrangement of interface demands a considerable speed penalty in comparison with a parallel interface connected directly to the computer bus. This reduction in data rate may be attributed to three factors:

(1) The use of RS232 serial communication, which limits the speed to 9600 baud.
(2) The sending of data in split format, so that a 22-bit word requires four bytes.
(3) The use of Instruction Bytes to initiate the transfer of data with each PIC.

Therefore, it can be concluded that the reading or writing of a 22-bit word requires six bytes to be transmitted along the serial lines. For a serial communication speed of 9600 baud, the resulting data speed would be 160 words per second, where each word is 22 bits in length.

6. MIGRATION OF INTELLIGENCE FROM THE PC TO THE PICS

The use of this interface to connect parallel-data sources to the host computer is possible in cases where the speed is not of great importance, such as the measurement of environmental variables including temperature, humidity and wind-speed, where sampling of the values is usually taken slowly. It is also relevant to industrial measurement and control situations where high speeds are not required.

However the major benefits of the interface accrue from the fact that the PICs can do more than just pass data. By suitable programming, they can be made to undertake some of the complex operations, which would normally be carried out by the computer. This is effectively a form of parallel processing, and frees the computer from tasks that would otherwise compete for time on the CPU.

For example, the intelligent interface has been programmed to act as a frequency counter, which converts the frequency into Hertz and communicates the resulting values to the computer. Another application is that of telephone monitoring, where the counting of dial pulses, detection of ringing tone, and duration measurement are all undertaken in the PICs, before transmission of the results to the computer.

Thus, the low speed of transmission is largely offset by the intelligence, which can be built into the interface itself.

7. CONCLUSIONS

An interface has been described which is cheap, light and highly portable, requiring connection only to the COM port of a personal computer in order to provide 22 lines of parallel bi-directional I/O. The hardware simplicity makes for ease of maintenance and robustness, and is particularly relevant in developing countries where import of sophisticated integrated circuits can be problematic.

Despite the low data rate, the interface is applicable to cases where slowly changing variables need to be communicated to a computer. Additionally the interface can be programmed to undertake processing of the data as it is passed between the computer and the external environment.

8. REFERENCES

1. Collier, M., “An Automatic Testing System for Electronic Party Line Subscriber Cards”, ***Transactions of South African Inst.of Electrical Engineers***, (March 1997), 1-7.
2. "PIC16/17 Microcont. D-Book", Microchip, Chandler, USA, (1995).