Tony Marchese AB2LX

General-Purpose Interface Board for the ISA Bus

A simple, inexpensive alternative for interfacing real-time, home-brewed applications to the PC.

Like many of you, I enjoy tinkering with electronic projects. Over the years, my projects have progressed from simple analog circuits to complex designs incorporating various forms of digital control. One of the most frustrating hurdles faced along the way was the lack of a simple method to test the digital interface.

A personal computer seemed to be the ideal candidate for the job. Unfortunately, interfacing to standard serial communication (COM) and parallel (LPT) ports was often frustrating considering the hardware and software constraints. The remainder of this article details the theory and construction of a general-purpose input/output (GPIO) card for IBM ISA/EISA bus-compatible systems. The GPIO provides a simple, inexpensive method for interfacing real time, home-brewed applications to the PC and is intended as an alternative to existing COM and LPT ports. Parts for the card, including the homemade printed circuit board, cost about $10. The GPIO provides all of the address decoding and bus buffering required to equip an ISA/EISA-compatible PC with 24, independently controllable external I/O lines.

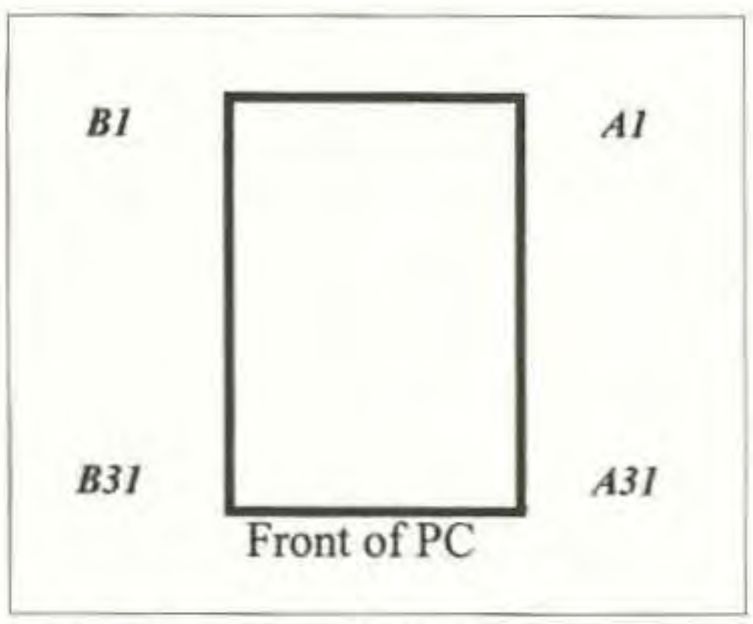

Fig. 1. ISA connector layout.

Technical background

The majority of IBM PC-, XT-, AT-compatible personal computers, with the exception of the IBM PS/2, utilize the "Industry Standard Architecture" (ISA) bus structure for the addition of feature cards. The ISA connector is detailed in **Fig. 1** and associated signals are defined in **Table 1**. The original PC and XT versions incorporated 8-bit processors. A single, dual-in-line, 62-pin connector is specified which provides access to 20 address lines, 8 data bits, power supplies, and control lines. The advent of the AT platform prompted an expansion of the original ISA bus, formally referred to as "Extended Industry Standard Architecture" (EISA), to accommodate the 16 data bits and 24 address lines of the 286 processor. Compatibility is maintained with existing ISA hardware by means of a secondary connector rather than changes to the original bus structure. Three minor changes were made, none of which significantly affect the structure.

Application boards connected to the IBM expansion bus should also comply with the following guidelines to prevent interference with other resident hardware:

1. No more than (2) TTL-LS loads attached to any given bus line.
2. NMOS, LSI components should be buffered from the connector as they can neither supply sufficient drive to the bus nor tolerate the negative spikes which may exist.
3. Boards must comply with predefined addressing.
4. Tri-state buffers should be utilized to prevent bus contention.

Theory of operation

The schematic for the GPIO card is provided in **Fig. 4**. The board incorporates U1 and U2, 74LS138, 3 to 8 decoders, to comply with standard IBM PC/XT/AT port allocations, while U3, an 74LS245 octal, tri-state transceiver, buffers the bidirectional data bus. The decoders utilize address lines A2–A9 to locate the general-purpose board at I/O port locations $300–$31F. IBM-

PIN	SIGNAL	DESCRIPTION
A2	D7	Bi-directional data bus - description relates to processor initiated bus cycle
A3	D6	
A4	D5	WRITE - processor places data on the bus prior to the rising edge of IOW or MEMW which
A5	D4	clocks the data into the port or latch
A6	D3	
A7	D2	READ - port or memory must place data on the bus prior to the rising edge of IOR or MEMR
A8	D1	which latches the data into the processor
A9	D0	
A12	A19	Address lines - 20 lines that provide 1Mbyte of memory addressing. Only lines A0-A15 are
...	....	used for port addressing
A31	A0	
		CONTROL SIGNALS
A1	I/O CH CK	I/O channel check - active low signal used to inform processor of parity error in I/O or memory
A10	I/O CH RDY	I/O channel ready - input to processor used to generate wait states by extending the length of bus cycles for slow memory or I/O
A11	AEN	Address Enable - signal asserted by the direct memory access (DMA) controller to indicate a DMA cycle is in progress. Typically used to disable I/O decoding such that DMA data is not inadvertently used as a port address.
B2	RESET DRV	Reset drive - used to initialize system logic during power up. This signal is synchronized with the falling edge of OSC
B11	MEMR	Memory read / write - active low signals used to control memory read & write operations
B12	MEMW	
B13	IOR	I/O read / write - active low signals used to control I/O port read & write operations
B14	IOW	
B30	OSC	System oscillator which provides a 70 ns (14.31818 Mhz) squarewave
B20	CLOCK	4.77 Mhz or 7.16 Mhz waveform depending upon system type
B28	ALE	Address Latch Enable - low to high transition indicates beginning of a processor initiated bus cycle. System bus does not contain valid address information when ALE is asserted. Valid address information is latch on the high to low transition of ALE.
		POWER SUPPLIES
B1,B10,B31	GND	system ground
B3, B29	+5V	
B5	-5V	
B7	-12V	
B9	+12V	
		INTERRUPTS
B4	IRQ2 (IRQ9)	Interrupt request - An interrupt is generated by asserting the IRQ line and holding it high until the processor acknowledges the request. The request is typically acknowledged in the interrupt service routine (ISR). The ISR may use the OUT command to set a I/O port bit which notifies the device to release the IRQ line. IRQ2 is the highest priority hardware interrupt available on the bus. IRQ2, pin B4, is replaced with IRQ9 in systems which utilized the EISA bus. The system's BIOS typically redirects the IRQ9 vector to that of IRQ2 to maintain compatibility.
B25	IRQ3	
B24	IRQ4	
B23	IRQ5	
B22	IRQ6	
B21	IRQ7	
		DMA CONTROL LINES
B18	DRQ1	DMA Request - synchronous channels used by peripheral to obtain DMA service. DMA request must remain high until the corresponding DACK line goes low. DRQ0 is not available on the bus as it is used to refresh the system's dynamic RAM.
B6	DRQ2	
B16	DRQ3	
B19	DACK0	DMA Acknowledge - active low signals used to acknowledge DMA request (DRQx). DACK0 is used to refresh dynamic RAM.
B17	DACK1	
B26	DACK2	
B15	DACK3	
B27	T/C	Terminal Count - provides a pulse when the terminal count for the DMA channel is reached

Table 1. *ISA bus pin definitions.*

compatible systems reserve these port addresses for prototype cards such as this. A bank of address-select jumpers is provided to allow the user to select from eight distinct 4-byte address blocks within the allocated space. The jumper configuration is shown in **Fig. 2.**

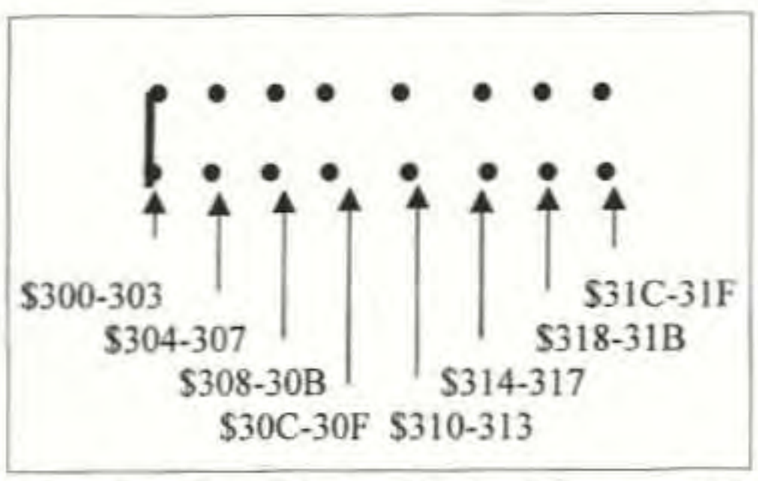

Fig. 2. *J3 address block (shown with jumper installed for $300-303).*

The decoder section also utilizes the Input/Output Read (IOR), Input/Output Write (IOW), Address Enable (AEN), and Address Latch Enable (ALE) lines from the system expansion bus. The IOR and IOW signals control the direction of read and

A1	A0	IOW	IOR	FUNCTION
0	0	1	0	read Port A data
0	0	0	1	write to Port A
0	1	1	0	read Port B data
0	1	0	1	write to Port B
1	0	1	0	read Port C data
1	0	0	1	write to Port C
1	1	1	0	undefined
1	1	0	1	write PPI control register

Table 2. *82C55 PPI internal registers.*

	40	20	
PC6	•	•	PC5
PC7	•	•	PC4
PA3	•	•	PC0
PA2	•	•	PC1
PA1	•	•	PC2
PA0	•	•	PC3
PA7	•	•	PB0
PA6	•	•	PB1
PA5	•	•	PB2
PA4	•	•	GND
+5V	•	•	GND
PB7	•	•	+5V
PB6	•	•	+5V
PB5	•	•	SPARE
PB4	•	•	SPARE
PB3	•	•	SPARE
+12V	•	•	SPARE
+12V	•	•	IRQ2(IRQ9)
-12V	•	•	-5V
-12V	•	•	-5V
	21	1	

Fig. 3. *J2 output connector (rear view).*

write operations. These IOR and IOW lines are active low signals which may be asserted by either the processor during port operations or the controller during direct memory access (DMA). The high-level AEN signal seen during DMA cycles inhibits the decoder, thereby preventing accidental interpretation of the data transferred during the process as a valid port address. The processor initiates a bus cycle by asserting and holding ALE until the address bus stabilizes. The ALE signal inhibits the decoder during this period to prevent activation during undefined processor bus states.

A valid port address activates the decoder. The decoder enables the chip-select (CS) on the 82C55 and the U3 data buffer. The general-purpose card also uses address lines A0 and A1 to select one of the four individual registers within the 82C55 programmable peripheral interface (PPI). The IOR/IOW lines and A0 and A1 can now provide access to the 82C55's internal registers as detailed in **Table 2**. A complete data sheet, including a description of the 82C55 registers, is available at [www.intersil.com/data/fn/fn2/fn2969/fn2969.pdf].

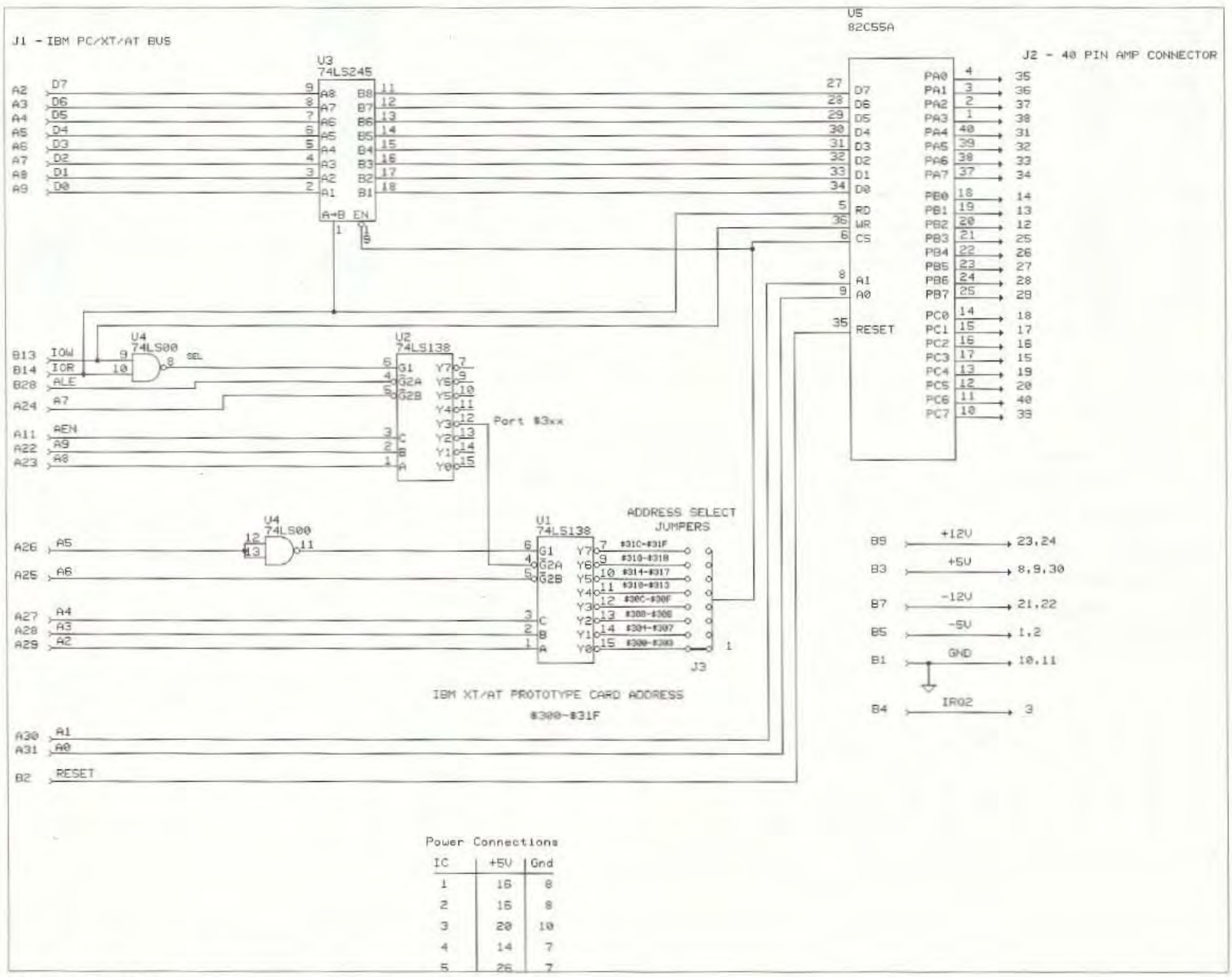

Fig. 4. *GPIO schematic.*

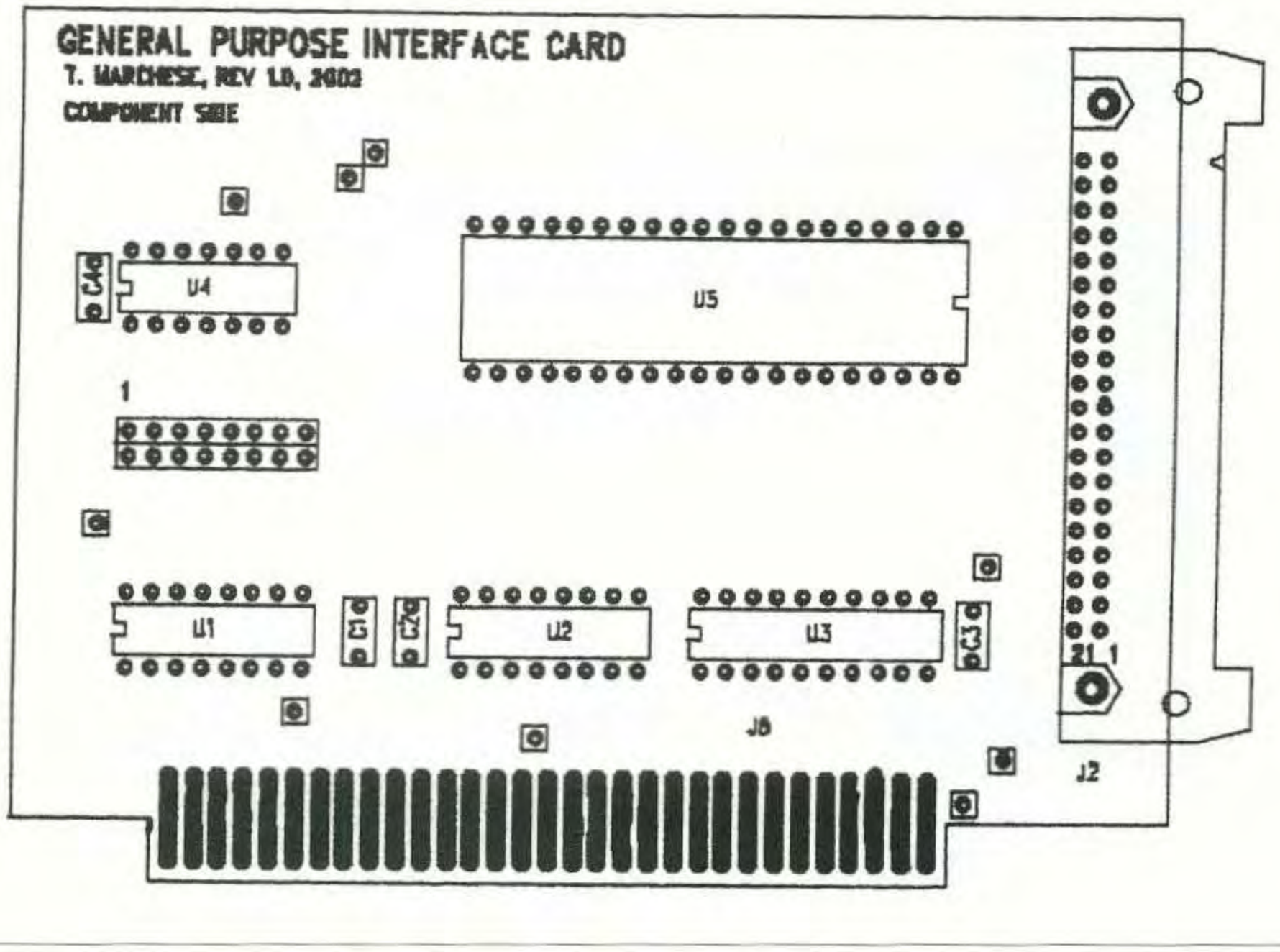

Fig. 5. GPIO PCB layout.

Construction

The circuit can be constructed on either a prototype card or with the printed circuit board (PCB) artwork provided in **Figs. 5, 6,** and **7**. The original interface card utilized a prototype board with a combination of wire-wrap and hard-wired construction. The card functioned well, but occupied the space of two ISA slots because of the long wire-wrap pins. The PCB is

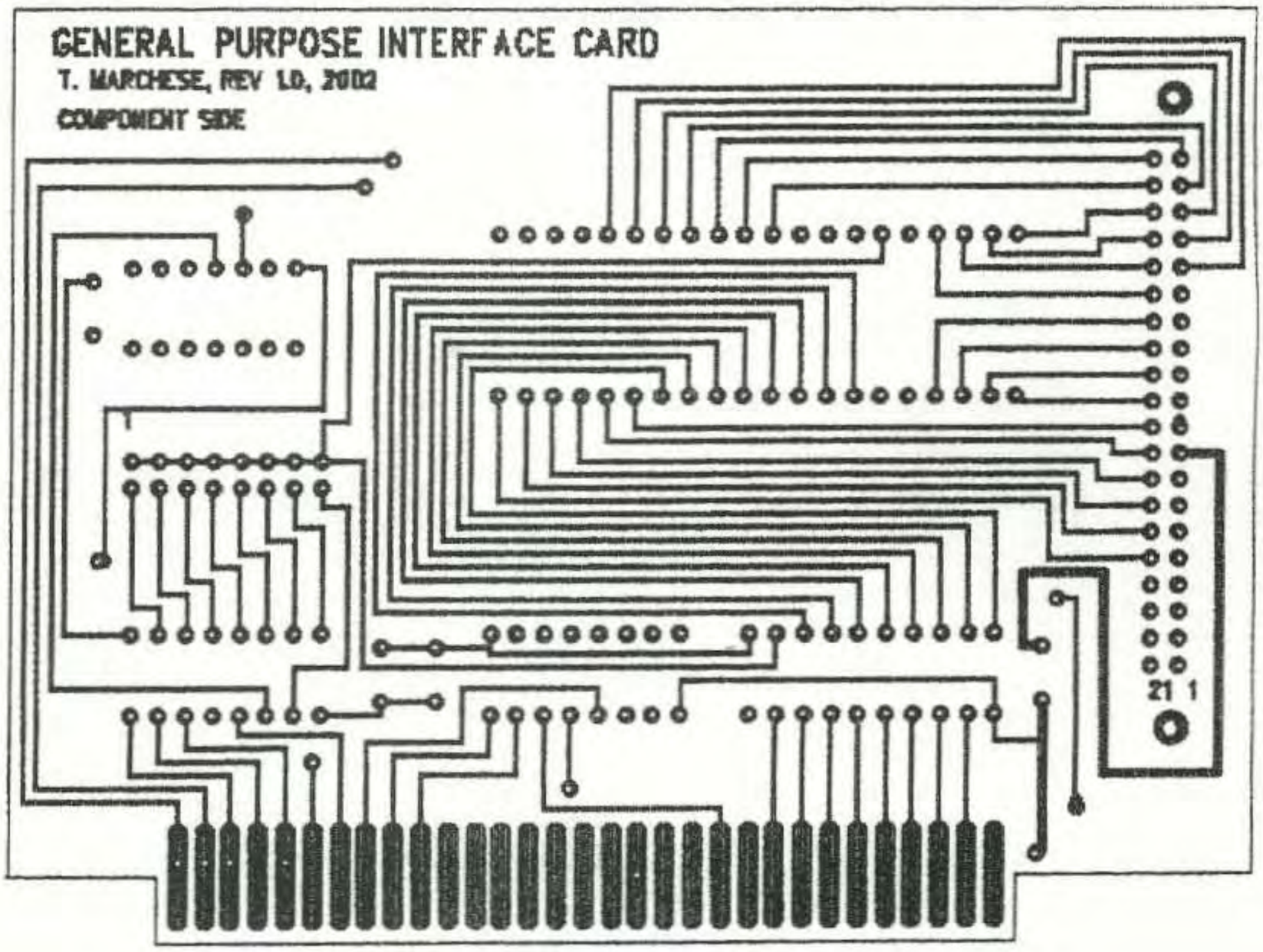

Fig. 6. GPIO PCB component side.

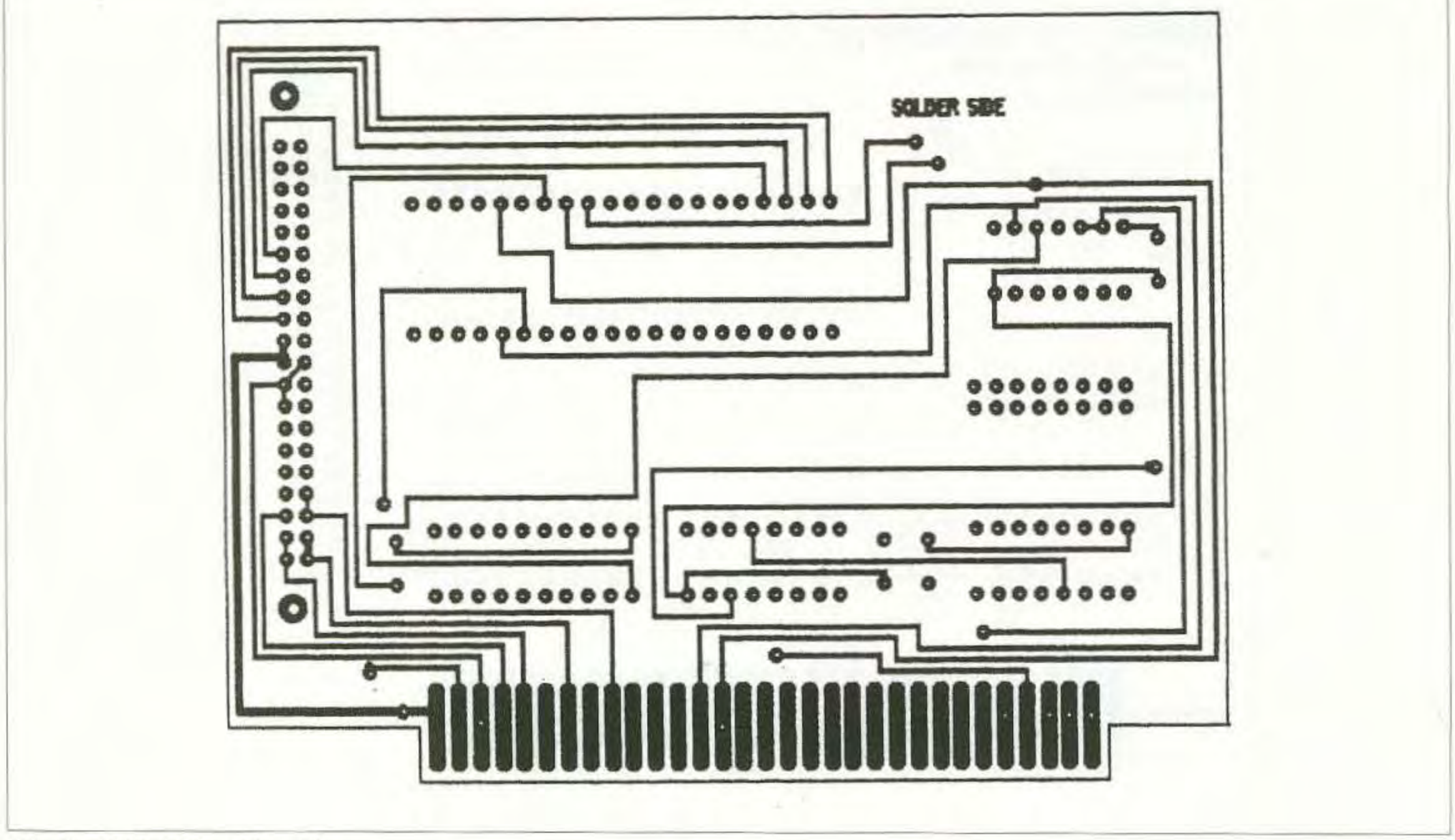

Fig. 7. GPIO PCB solder side.

provided to simplify construction and minimize the space requirements as the card fits in a single, short slot of the ISA bus.

The circuit board includes provisions for J2, a 40-pin connector which matches up nicely with ribbon cable. The J2 connector and cable allows easy access to all of the 82C55's 24 input/output lines. The ±5V, ±12V, and ground lines from the bus are also made available to the user on J2.

The printed traces are relatively thin. Current drawn from these lines should be limited to less than 20 mA. Extreme caution must be exercised when using these lines as they are tied directly to the computer system's power supply. Remember to protect the computer from damage by adding fuses to the power supply lines when they are used by an external application.

Fuses are purposely not included on the GPIO board for several reasons. First, the fuse should be appropriate for the application. Incorporating a large fuse to protect the system's power supply will undoubtedly result in damage to your application board if a fault occurs. Second, accessing an external fuse is typically easier than opening the computer and removing the GPIO card. This sounds good on paper, but I'll let you know if I damage my first GPIO card.

IRQ2 is also available on J2, as real-time applications often use a hardware interrupt. Please note that this interrupt is mapped to IRQ9 in EISA bus systems.

A test routine, written in QBasic, is provided here for assessing the card's functionality after construction is complete. The program simultaneously toggles each of the 24 I/O pins. Testing is straightforward: Run the program, then measure any of the PA(x), PB(x), or PC(x) pins with a voltmeter. The output should toggle between less than 0.5V to greater than +4.5V. Save this program, as it will also facilitate testing of the card if problems occur at a later time.

Name	Type	Description	Cost	Jameco P/N
U1	74LS138	3 to 8 decoder/demultiplexer	$0.29	46607
U2	74LS138	3 to 8 decoder/demultiplexer	$0.29	46607
U3	74LS245	Octal bus transceiver, tristate	$0.39	47212
U4	74LS00	Quad 2-input NAND gate	$0.19	46252
U5	82C55	Programmable peripheral interface	$4.95	52425
C1–C5	—	0.1 µF mylar cap	—	—
J2	—	40-pin header	$0.55	53604
—	—	Printed circuit board	—	—

Source: Jameco Electronics, 1355 Shoreway Rd., Belmont CA 94002-4100; 800-831-4141; [www.jameco.com]

Table 3. Parts list.

```
'*************************************************

' Test program for General Purpose Interface Card
' Program toggles all PPI bits between high & low
' state to facilitate testing.  The address jumper, J2
' is assumed to be in the $300-303 position.
'
' Description            Hex     Decimal
'                        Address Address
'
' Port A Read / Write     $300    768
' Port B Read / Write     $301    769
' Port C Read / Write     $302    770
' PPI Control Register    $303    771
'
'    Written by Tony Marchese, AB2LX, 2002

DECLARE SUB timep (x)

Rcontrol = 771

' numpulses may be increased  if a longer test time is required

numpulses = 10
high = 255
low = 0

'set PPI to Mode 0 which configures all interface lines as outputs

OUT Rcontrol, 128

' set up to output (qty) of pulses
FOR qty = 1 TO numpulses

' set all output bits to the high state

portstate = high
CLS
PRINT "Port Bits = 1 "
PRINT "Pulse Number: "; qty
timep (portstate)

'clr all output bits to the low state

portstate = low
CLS
PRINT "Port Bits = 0 "
PRINT "Pulse Number: "; qty
timep (portstate)

NEXT qty

END

SUB timep (portstate)

PortA = 768
PortB = 769
PortC = 770

' time delay routine holds output pulse in portstate for time = duration

FOR duration = 1 TO 20000

OUT PortA, portstate
OUT PortB, portstate
OUT PortC, portstate

NEXT duration

END SUB
'*************************************************
```

Test program.

Conclusions

The GPIO has come in handy for several projects, including as an interface to a stand-alone LCD matrix. The LCD required special timing and control signals which were not readily available from the parallel port. The GPIO provided an easy method to investigate the interface requirements, as the control signals and sequences to the display could be easily manipulated using relatively short QBasic routines. 73