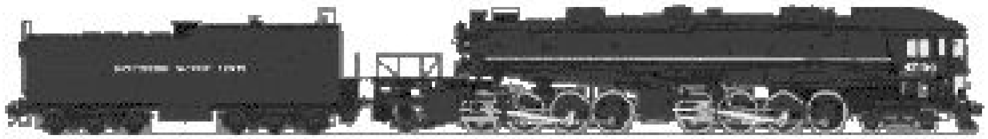


THE COMPUTER/ MODEL RAILROAD INTERFACE (C/MRI)

USER'S MANUAL
Version 3.1a

By Dr. Bruce A. Chubb, MMR

Chapter 6: Testing Serial-Based Nodes

www.jlcenterprises.net

©2013 Bruce A. Chubb

Chapter 6

Testing Serial-Based Nodes

Assuming you built up your own serial based node (using an SMINI or an SUSIC) it is important to test what you have built before putting it to use. This chapter will show how to test your cards to verify that everything is assembled and connected correctly. To help with the testing we will use special test software loaded into your PC.

Even if you purchased your card(s) assembled and tested, it is still a good idea to carry out the testing. It helps in becoming familiar with C/MRI node operation, it checks that your interface connections to the computer are correct and verifies that nothing was damaged during shipping.

The methods used to test and debug an SMINI node and an SUSIC-based node are very similar so I have combined the material into this single comprehensive chapter. Where the testing and debug steps are different between the two node types, I cover both and you can skip the portion that does not apply to your immediate need.

Testing is significantly automated using a Universal Serial Test Program. Two versions are provided on disk:

1. USTPQB.BAS for use with QuickBASIC operating under DOS
2. USTPVB.VBP for use with Visual Basic operating under Windows

Both programs provide an identical level of testing.

To perform the testing you need a standard test card and wraparound cable. If you already have a previous DOTEST card and cable you can use them for SMINI testing. However starting from scratch, and especially if you use the new 32-bit I/O cards with the SUSIC, I recommend making use of the new TEST32 card. The advantage of the 32-bit card is its ability to test the new DOUT32 card as well as previous 24-bit cards and also the SMINI.

Throughout this chapter, for ease of example, I will reference all I/O cards as being the new 32-bit cards (DOUT32 and DIN32). Be assured however that all the tests work just as well if you are using any of the 24-bit cards (DOUT, DIN, COUT24 and CIN24). Similarly, I will be referencing mainly the SUSIC serial card. However, all tests work just as well using the original USIC design with the 68701 Microcontroller. Fundamentally, all the tests presented in this chapter are truly Universal Serial Tests and they work equally well with all the different C/MRI serial cards.

ASSEMBLING THE 32-BIT OUTPUT TEST CARD (TEST32)

We will soon be using software to test our SMINI or SUSIC-based nodes. The first test checks that messages can be sent from the PC to the output pins on the SMINI and the DOUT32. To do that efficiently we need a test card. Then, once all the outputs test satisfactorily, we can use the test card combined with a wraparound test cable to test reading the SMINI and DIN32 inputs.

Fig. 6-1 is an illustration of the TEST32 card. The LEDs L1 through L32 are used to display the output status of a DOUT32 card, the 24 outputs on a DOUT card or the two sets of 24 outputs on an SMINI.

This greatly simplifies the testing of outputs and is very useful any time you need to debug the C/MRI or confirm that it is operating properly.

Only 24 of the 32 LEDs, marked as Ports A through C are used when testing the SMINI and DOUT cards but the others are required for testing DOUT32 cards.

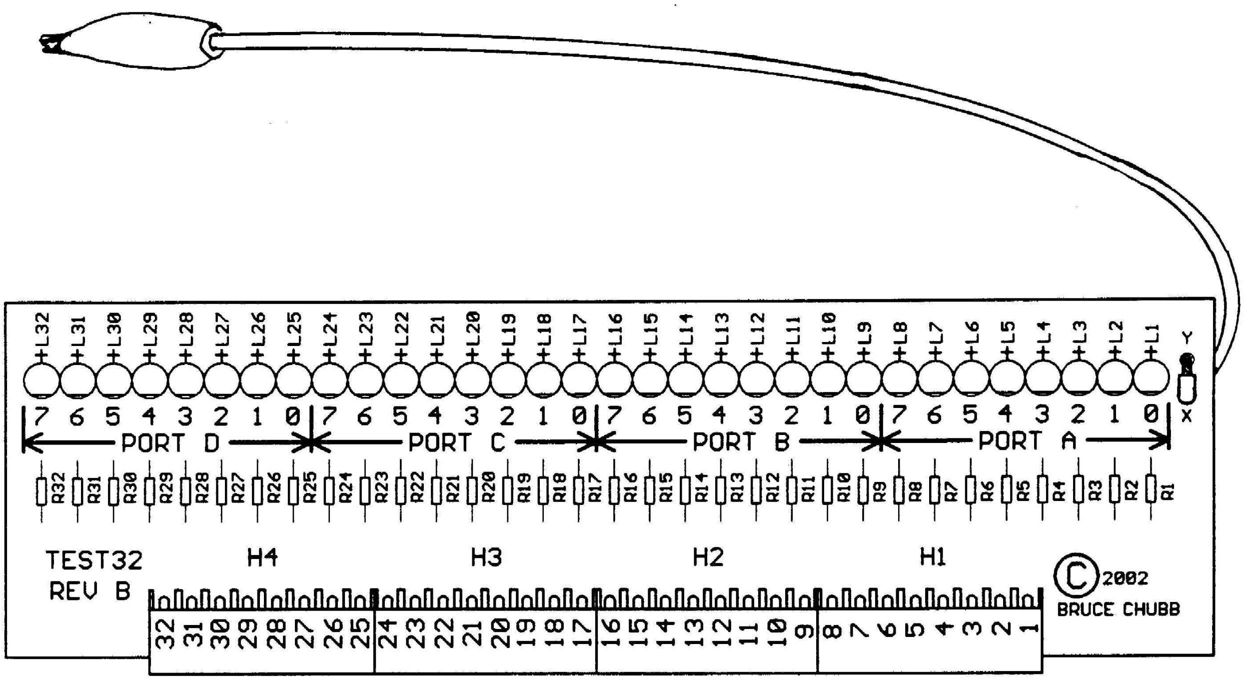


Fig. 6-1. TEST32 card parts layout

A ready-to-assemble TEST32 printed circuit card is available from JLC Enterprises and a complete kit including 32 bicolor LEDs is available from SLIQ Electronics, in addition to fully assembled and tested cards. See Appendix A for details. For do-it-yourselfers, Table 6-1 lists the required parts.

Table 6-1. TEST32 card parts list

Qnty.	Symbol	Description
32	R1-R32	330Ω resistors [orange-orange-brown]
4	H1-H4	8-pin Waldom side entry connectors (Mouser 538-09-52-3081)
32	L1-L32	Bicolor diffused red/green T1 size 2-lead LEDs (Mouser 604-WP93EWG) or if only using to test current sinking outputs can use diffused T1 size Red LEDs (Jameco 333851) or if only using to test current sourcing outputs can use diffused T1 size Green LEDs (Jameco 333201)
1	—	Clip lead (use Jameco 10444 or any equivalent)

Author's recommendations for suppliers given in parentheses above with part numbers where applicable. Equivalent parts may be substituted. Resistors are ¼W, 5 percent and color codes are given in brackets.

In general, I recommend using the 2-lead bicolor LEDs as this enables testing both current-sinking and current-sourcing outputs. Connect the clip lead to +5Vdc when checking standard current-sinking outputs and to ground when checking alternate current-sourcing outputs.

However, bicolor LEDs are more expensive than single color LEDs. Therefore, if you plan only to use one type of output (standard current-sinking or alternate current-sourcing) you can install the less expensive single color LEDs. I like to use the red LED for testing standard current-sinking and the green LED for testing alternate current-sourcing outputs. Using this procedure it is easy to tell the cards intended function plus the colors match those that light when using the multi-purpose bicolor LEDs.

R1-R32.

****Important Point****

If unclear as to which lead is which with the LEDs you have purchased, I strongly recommend that you install only the first LED and once you have successfully passed the card test for that LED, install the remaining LEDs. Having all the LEDs in backwards when you try to test is quite undesirable – especially when they all need to be unsoldered and then reinstalled correctly.

L1-L32[+]. Using the specified bicolor LED, install the longer lead in the hole with the square pad. This should provide a red indication when checking standard current-sinking outputs and a green indication when checking the alternative current-sourcing outputs.

If using the red only LED, install the longer lead in the hole with the + sign and square pad and the test card works only for testing standard current-sinking outputs.

Similarly, if using the green only LED, install the shorter lead in the hole with the + sign and square pad and the test card work only for testing the alternative current-sourcing outputs.

H1-H4. Install the 8-contact side entry connectors by first hooking their nylon retaining fingers over the edge of the card, then feeding the metal contact pins through the card holes. Make sure all 8 pins of each connector pass through the holes. Hold the connector shell tightly against the card as you solder.

Clip lead. Cut off and discard one of the alligator clips, making the remaining lead as long a possible. Enlarge hole “x” to be a snug fit for your clip lead’s insulation diameter and hole “y” to fit the clip leads internal wire diameter. Feed the cutoff end of the clip lead up from the bottom side of the board through hole X, strip off about ¼ inch of the insulation and then fit the stripped end into hole “y”. Solder and trim. Then work all slack back toward the clip.

Cleanup and inspection.

That is all there is to assembling the Test32 Card.

TEST32 CARD TESTING

Before using the TEST32 card, it is important to make sure it is functioning correctly.

If using the bicolor LED, connect the card's clip lead to +5Vdc. Use another clip lead to step through touching each of the connector pins on H1 through H4 to ground. Each time the corresponding LED should light red. Then connect the test card's clip lead to ground and use the test clip lead to step through touching each of the connector pins on H1 through H4 to +5Vdc. Each time the corresponding LED should light green.

If you are using the red only LED, connect the card's clip lead to +5Vdc. Use a second clip lead to step through touching each of the connector pins on H1 through H4 to ground. Each time the corresponding LED should light red.

If you are using the green only LED, connect the card's clip lead to ground. Use a second clip lead to step through touching each of the connector pins on H1 through H4 to +5Vdc. Each time the corresponding LED should light green.

That is all there is to testing the TEST32 card. We need to assemble one more item, a special cable, to be ready for system testing.

ASSEMBLING WRAPAROUND TEST CABLE

For system testing we will be wrapping the outputs from our SMINI (or from a DOUT32 card) back in as inputs to our SMINI (or to a DIN32 input card). For that we need to construct the wraparound test cable shown in Fig. 6-2.

Each pin of one connector is simply wired straight to the corresponding pin of the other connector. Fig. 6-2a shows the typical arrangement to create the 32-wire cable using 4 identical 8-wire sections.

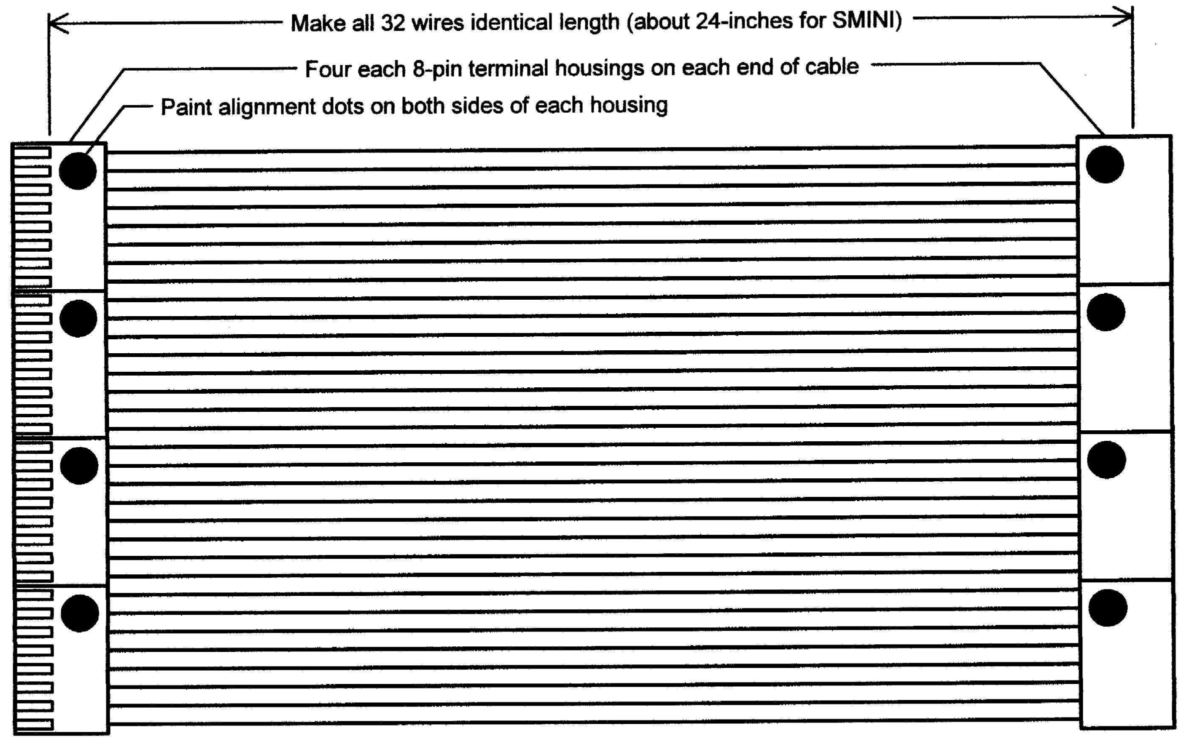
However if you already happen to have a 24-wire cable from previous C/MRI testing, and you need 32 wires, simply assemble a single 8-wire section shown in Fig. 6-2b. Although it is seldom listed in distributor catalogs, Molex (Waldom) does make a single 24-pin housing.

Table 6-2 lists the required parts for making the wraparound cable.

Table 6-2. Wraparound cable parts list

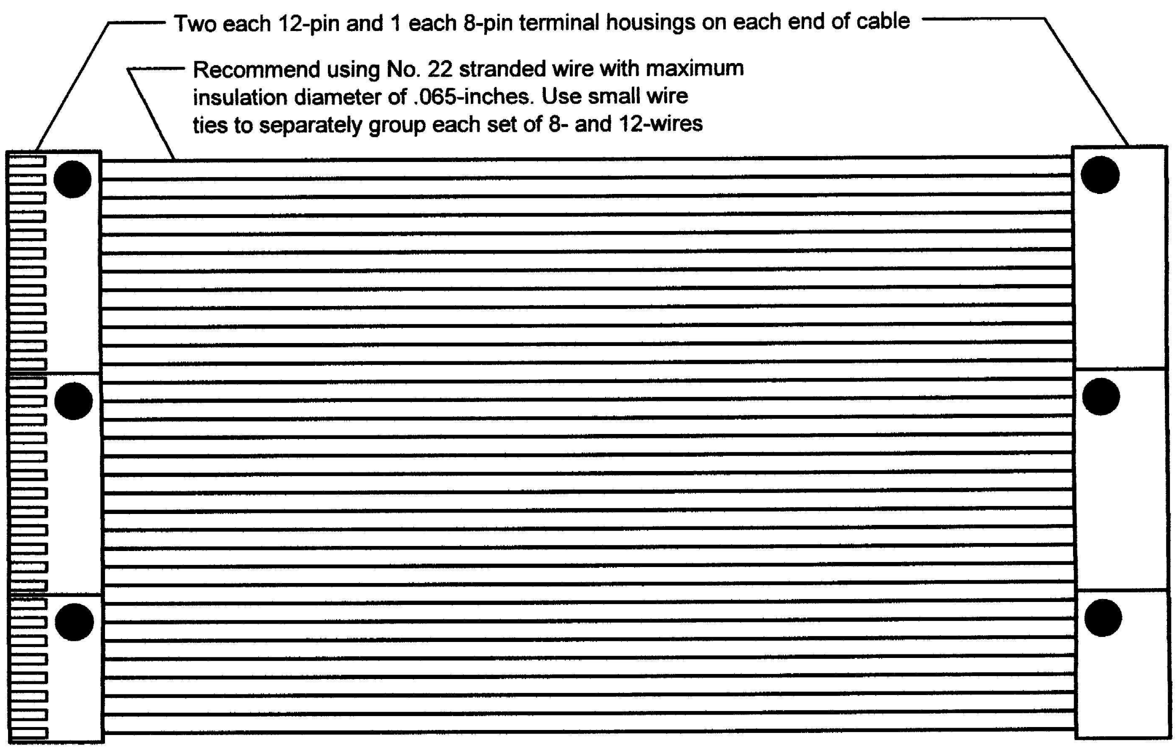
Part description	Recommended source and part number
8-pin Waldom terminal housing	Mouser 538-09-50-3081
12-pin Waldom terminal housing	Mouser 538-09-50-3121
Crimp terminals for wire sizes 18-24	Mouser 538-08-50-0106
Crimp terminals for wire sizes 22-26	Mouser 538-08-50-0108
Stranded insulated hook-up wire	Your choice, see note in Fig. 6-2

Cut all 32-wires the same length. To make it easy to connect to both ends of the SMINI card, the best length is about 24 inches. Easy and reliable connections require using the Special Crimping Tool (Digi-Key part number WM9903) as demonstrated in Fig. 3-11. Do not try to use a general-purpose crimping tool, as they are not designed to wrap the tabs around the wire and around the insulation. You will not form good connections.



a) Approach using 8 each 8-pin housings

Note: Drawing not to scale.
Cable appears shorter relative
to size of connector housings.



b) Approach using 4 each 12-pin and 2 each 8-pin housings

Fig. 6-2. Wraparound test cable

Additionally, the number of I/O cables required for even a small interface makes the $15 tool cost a wise investment. Once you become familiar with using this special tool, crimping is easy and fast. Also, because the connections made by the tool are so tight, you can typically skip the soldering requirement. However, for added reliability, we do solder each of the crimps on the SVOS.

I recommend using AWG 22 stranded wire with an insulation diameter less than .065 inches. Wire with thicker insulation does not fit well into the housings. Stranded wire is essential because of the repeated flexing each time the cable is attached and removed. Assembling the cable with the tab locking notches facing up on one end and down on the opposite end makes for identical cable attachment when plugged onto the two cards. This arrangement is illustrated in Fig. 6-2.

Also, it is important to paint a colored dot on each cable end, on both sides of the connector housing, denoting the same contiguous wire. This helps to insure having the cable plugged onto the cards correctly. Believe me, I have helped, over the telephone, many a first time user achieve success in their wraparound test merely by correcting a twist in their wraparound test cable.

To avoid cable misconnections, use the colored dots to check that you have the same cable alignment for each card to which the cables are attached. Also, once you have your dots painted, it is desirable to attach a few small wire-ties along each cable section. Such action makes the cabling neater and helps prevent having the 32 separate wires becoming tangled.

Before beginning actual testing, it is important to be in position to take advantage of the status information conveyed by the green, yellow and red LEDs mounted on each SMINI and SUSIC.

USING THE STATUS MONITORING LEDS (SMINI and SUSIC)

Both the SMINI and the SUSIC, as long as they are powered up, continuously blink a green status LED to tell you how the card is operating. Different blink patterns indicate if the PIC16F877 is operating in its idle mode, receiving data without errors and if encountering errors what type of errors are present. Learning to recognize these important blinking patterns can help you in testing and debugging a system. Just as important, the green LED tells you if the SMINIs and SUSICs are correctly initialized and functioning without detected errors. Table 6-3 summarizes the 7 different blinking patterns programmed into the PIC16F877.

These codes are significantly expanded beyond the 3 codes (idle, no errors and errors) provided with the original USIC. Subdividing the error code into five different error types proves very helpful in cases where you might run into problems. It is important to note that LED Modes 0 and 1 depict normal operation, while Modes 2 through 4 denote general error situations that can occur anytime. By contrast, Modes 5 and 6 occur only when invalid user specified initialization data is sent to the PIC16F877.

Watching for these seven different indications during C/MRI operation can help to get your system initially operating, then help you to locate and solve future problems should they develop.

When an SMINI or an SUSIC is first powered up, and prior to turning on your PC, the green LED should always be in Mode 0. This is the idle mode for both the SMINI and SUSIC. It indicates that the PIC16F877 is powered up correctly and that its internal program is functioning and it is waiting to receive initialization inputs from the PC. Per Table 6-3, the blink pattern for this idle mode is 1.8 seconds on and .2 seconds off.

Table 6-3. Blink modes for the green status LED on the SMINI and SUSIC

LED MODE	Microcontroller Operational Status	On Time (seconds)	Off Time (seconds)
0	Idle mode (877 powered up OK but not yet initialized from the PC)	1.8	.20
1	Initialized OK and 877 receiving data from PC without any detected errors	1.0	1.0
2	Receiving data from PC with framing errors; typically occurs as a result of incompatible baud rate settings	.05	.05
3	Receiving data from PC with overrun errors; occurs when 877 is not otherwise keeping up with flow of input bytes being received from the PC	.05	.20
4	Receiving data with DLE or ETX errors; typically occurs when message from PC is garbled due to transmission line noise or message formatting problems	.05	1.0
5	Message type or node definition parameter error during initialization; typically occurs when sending bad initialization data	.20	.20
6	NS out of range and/or invalid CT data during initialization; typically occurs when NS and CT array elements are defined incorrectly	.20	1.0

When the PC program sends initialization data to the PIC16F877, you should see a quick burst of activity on the yellow, receiving data, LED. Correspondingly, as soon as the 877 is properly initialized, the green LED blink rate changes from Mode 0 to Mode 1 – receiving data without errors – with a corresponding blink rate of 1 second on and 1 second off.

When everything continues to operate correctly, the 1 second on and 1 second off blink rate should continue all the time your railroad is operating. In addition, the yellow LED blinks anytime the PIC16F877 is receiving data from the PC and the red LED blinks anytime the PIC16F877 is sending data to the PC.

Typically in multi-node system applications the yellow LED will be considerably more active than the red. This is because all nodes receive all the data being sent. By contrast, the red LED activity for any given node is much less because it is generated only when the addressed node is sending data to the PC.

During normal operation the green status LED should not deviate from the Mode 1, 1 second on and 1 second off rate, unless for some reason the PIC16F877 detects an error. The type of error detected is indicated by Modes 2 through 6 in Table 6-3. For example, setting a node's SW2 DIP switch for 9600 baud when the PC program is set for 19200 baud typically generates Mode 2 flashing – receiving data with framing errors. This super-fast flash rate – .05 seconds on and .05 seconds off – is very easy to spot. You will most often see this flash pattern only for the condition of incorrectly set baud rates.

Because the PIC16F877 program operates extremely fast and typically has no trouble keeping up with your PC, which generally performs much more processing than does the 877, you may never observe Mode 3. However, if the 877 does run into the situation where it cannot keep up with the rate at which the PC is sending data, the 877 displays error Mode 3 – receiving data with overrun errors.

Per Table 6-3 for Mode 3 the corresponding flash pattern is .05 seconds on and .2 seconds off. If this occurs, you typically need to add a delay to slow down your PC application program, or else to operate at a slower baud rate.

Every serial transmission system is prone to an occasional transmission error. How frequently these occur depends upon such factors as cable length, quality of cable, line termination, how electrically noisy is the environment, and the baud rate. When such errors are detected, the PIC16F877 program switches to LED Mode 4 – receiving DLE or ETX errors. The corresponding blink pattern, per Table 6-3, is .05 seconds on and 1 second off. Usually if this situation occurs it is for only a single data byte and the following data byte will be correct. Therefore, the occurrence is so short lived that you most likely will not observe the error occurring from the 877. By contrast however, when the PC detects these errors the standard PC Serial Protocol Subroutines can print out such errors. This printout is the best way to monitor the number of transmission errors occurring during an operating period.

The LED Modes 5 and 6 should only occur during or immediately after node initialization. They indicate bad user defined data included in the initialization message. For example, Mode 5 results if your application software defines a node type that was neither SUSIC nor SMINI. Mode 6 occurs if your application software tries to define a node with more than the maximum allowed 64 I/O cards, or an invalid card type.

Typically all these initialization-type errors are detected first by the improved standard Serial Protocol Subroutine packages introduced with V3.0 C/MRI User's Manual. The advantage of this approach is that the PC program prints out a more definitive diagnostic error message. However, if the particular Serial Protocol Subroutines you elect to use do not provide this diagnostic capability, any initialization errors generated by the PC program are likely to be picked up by the flashing green status LED on the SMINI or SUSIC.

There is one other point worth mentioning. If you happen to have your C/MRI powered up (i.e. your SMINI, or SUSIC is operating in its idle mode, 1.8 seconds on and .2 seconds off) and then you power up your computer, most likely you will observe the green LED changing to one of its error mode blink rates. This occurs when the PC's internal power-up testing routines check the RS232 port, using data not recognized by the SMINI or SUSIC. Simply ignore this situation because as soon as your application program sends a valid initialization message everything recovers for correct operation.

In summary, getting to know and understand the green, yellow and red LEDs blink patterns can help significantly in monitoring the health of your C/MRI and how to go about fixing problems should they occur.

USING THE STATUS MONITORING LEDS (Classic USIC)

The Classic USIC, similarly to the SUSIC and the SMINI, as long as it is powered up, continuously blinks its green status LED to tell you how the card is operating. Different blink patterns tell you if the MC68701 is operating in its idle mode, receiving data without errors or receiving data with detected errors. Table 6-4 summarizes the 3 LED modes with the USIC.

A major limitation of the USIC is that it only indicates if there are detected errors without showing the type of error. Fortunately, the improved diagnostic printouts provided by the new Serial Protocol Subroutines generally provide all that is required to isolate and solve communication problems.

Table 6-4. Blink modes for the green status LED on the Classic USIC (card using 68701)

LED MODE	Microcontroller Operational Status	On Time (seconds)	Off Time (seconds)
0	Idle mode (68701 powered up OK but not receiving data)	1.8	.20
1	Initialized OK and 68701 receiving data from PC without any detected errors	1.0	1.0
2	Receiving data from PC with detected errors	.05	.05

There is one subtle difference in how Mode 0 works with the Classic USIC, compared with the SUSIC/SMINI. All 3 cards power up in Mode 0. When correctly initialized the mode switches from 0 to 1 on all three cards. During normal operation, Mode 1 stays active on all 3 cards. However, with the Classic USIC anytime the data stream is stopped for a 10 second period, the green LED returns to its idle Mode 0 while the SUSIC/SMINI green LED retains its Mode 1 status.

Operation of the yellow and red LEDs, as explained in the previous section, is the same for the USIC, SUSIC and SMINI.

The remaining sections of this chapter concentrate on testing the SUSIC and SMINI based nodes. Where the testing is SUSIC-based, it assumes that 32-bit I/O cards (DIN32 and DOUT32) are used. The general techniques are fully applicable to the Classic USIC-based nodes and to using 24-bit I/O cards with the SUSIC. However some of the details regarding debugging, IC pin voltage measurements, and others need to be altered to fit the particular card/node configurations. For example, for the IC pin voltage measurements, appropriate tables are included within the appropriate chapters/appendices describing the different cards.

STANDALONE TESTING

My previous writings included conducting a lengthy set of standalone card tests prior to the application of test software. These pre-tests were designed to help establish correct card trace and cable connections, to verify correct parts insertion, to locate solder bridges and to ensure that each IC was properly powered.

Because almost all of the new C/MRI boards are factory tested for short and open circuits, because they include solder masking to minimize solder bridges during assembly, and overall there are fewer parts, I have eliminated most of the recommended pre-testing. Most users, performing careful assembly, will find their new C/MRI cards work perfectly the first time they apply power.

However, for users that discover they have a problem, I have greatly expanded the ***In Case of Difficulty*** material. This includes all the original testing plus much more to help in locating and solving problems. I do still recommend a few initial standalone tests prior to proceeding into using the power of software to verify correct operation of your entire C/MRI setup.

The most likely sources of trouble, if you have any, will be poor soldering and incorrect parts insertion. The next thing to look for is cable trouble between the C/MRI hardware and the serial port on your computer, or the way in which the Serial Protocol Subroutines interact with your computer's serial I/O port. Systematic checking takes time, but it goes a long way toward ensuring that your C/MRI operates correctly.

The standalone testing for the SMINI and the SUSIC based node is straightforward and is mainly to ensure that correct power is reaching the cards and that the Microcontroller, the PIC16F877, powers up correctly. To do so, check each of the following boxes only after you complete the requested action and, where applicable, get the correct results.

Check type of power supply. Make certain that your +5Vdc Power Supply is REGULATED. Do not use a conventional RR power pack set for 5 volts. Do not use a Wall-Mart or Radio Shack plug-in-the-wall type power supply unless it specifically states that the +5Vdc is a REGULATED output. Regulated means that the output voltage stays constant, in this case at +5Vdc, independent of the load, up to the rated supply current.

Check DC voltage level. Set your VOM to read DC voltage. Use a clip lead to connect the meter's black negative lead to the ground output of your power supply. With your power supply turned on, touch the meter's red positive lead to the power supply output labeled +5Vdc. The meter should read almost exactly 5Vdc. The maximum acceptable tolerance range is from 4.8 to 5.2Vdc.

Verifying supply regulation. This step is optional but highly recommended if you are unsure about the regulation capability of your supply. To verify that your supply is adequately regulated, connect a 3Ω resistor across the terminals of your supply. The resistor should have a power rating of at least 10W.

If properly regulated, your supply voltage should stay constant at the 5Vdc level independent of whether the resistor is attached or not attached. A very small change, say.05Vdc drop with the resistor attached is acceptable but larger changes, especially any change that causes the output to move outside the 4.8 to 5.2Vdc range, indicates an unacceptable power supply.

****Important point****

The SW1 and SW2 DIP switch settings are only read during the power-up cycle for the SMINI and the SUSIC. Therefore, if either of these switch settings is changed anytime while power is applied, the new settings will not take effect until the power to the card is cycled. Depending upon the power supply you have chosen, you may need to wait a significant period of time until its output voltage decays to zero before reapplying power. Adding a resistor load to your supply (typically referred to as a bleed-off resistor) can decrease this waiting time.

Connect power wiring. Turn off your power supply and use spade lugs to connect the ground output from your supply to the ground terminal screw on SMINI, or an IOMBX if you are using the SUSIC. Then make the similar connection for the +5Vdc. On an IOMBX the ground connection needs to be made to the terminal screw attached to the wide circuit trace along the card edge and the + 5Vdc connection to the terminal screw attached to the narrow circuit trace. Make absolutely certain that your connections are correct as a reverse polarity connection will most likely destroy most if not all the ICs, including the U1B Microcontroller and all polarity sensitive capacitors.

SW1, SW2. Set all segments on both the USIC address and baud rate DIP switches to off, corresponding to USIC address = 0 and baud rate = 9600.

Plug SUSIC into IOMBX for SUSIC applications. If you are testing an SUSIC-based node, plug the SUSIC into the first card slot of the IOMBX. Make certain that the component side of the SUSIC faces the direction indicated by the direction arrow pasted on the IOMBX. This ensures that the two adjacent ground pins on the IOMBX plug into H4 connector openings 1 and 2 on the SUSIC.

Turn on power. Plug in, or otherwise turn on your power supply, and measure the voltage across the two power terminal screws. A reading maintained at +5Vdc is acceptable. If the reading drops substantially toward zero it is likely that your supply isn't sufficiently regulated or there is a short in your SMINI, IOMBX or SUSIC. If locating the short is not obvious, I suggest temporarily jumping ahead to the ***In Case of Difficulty*** section. Then, once the problem is corrected, come back to this point to continue testing.

Idle operation test. Assuming you read +5Vdc, the green status LED on the SMINI or SUSIC should start blinking in its idle mode, 1.8 second on and .2 second off. It is controlled by a real-time loop counter in the PIC16F877's internal software and shows that the chip is looping properly. In most cases, the red and yellow LEDs should be off. Independent of whether the red and yellow LEDs are on or off, you may proceed to the next test. However, if the green LED is not blinking correctly, you need to stop at this point and fix the problem. For assistance, temporarily jump ahead to the ***In Case of Difficulty*** section. Then, once you get the green LED's idle mode working correctly, come back to this point.

> ****Important Point****
> Some power supplies with large filter capacitance can take several minutes or longer to decay to zero. If you cycle your power supply off and on and the green LED on the SMINI or SUSIC doesn't reset to its idle mode, this most likely means that your power supply takes a longer time to decay to zero. If so, you can obviate this problem by adding a toggle switch between the supply output and your C/MRI cards for use in cycling the power.

Computer connections. If you have equipped your SMINI or SUSIC for RS232 then connect it to the RS232 serial port of your computer using the details noted in Fig. 4-9a or b and the associated text. If you have equipped your SMINI or SUSIC for RS485 only, then you need to connect the RS485 Conversion Card as denoted by Fig. 4-9c or d and the associated text. Picking the a or b subfigure is dependent upon whether your particular PC's RS232 serial port uses a 25-pin or a 9-pin connector. Also, if you are using the RS485 card it needs +5Vdc and ground connections to your power supply.

USING QUICKBASIC AND VISUAL BASIC UNIVERSAL SERIAL TEST PROGRAMS

With the standalone testing complete, we are now ready to perform automated node testing using software. As mentioned at the beginning of this chapter, this can be accomplished using either QuickBASIC or Visual Basic. I will primarily focus on using QuickBASIC with the USTPQB.BAS program. This is the program I recommend using for initial C/MRI node testing. However, if you are already familiar with using Visual Basic, then feel free to use USTPVB.VBP, where you substitute your VB knowledge in place of the instructions I provide on using QuickBASIC and USTPQB.BAS.

For example, where I say "Bring up QuickBASIC", you simply "Bring up Visual Basic." Where I say "Open USTPQB.BAS", you simply "Open Project USTPVB.VBP." Where I say "Run USTPQB.BAS", you run "USTPVB.VBP. Both programs use extensive prompting to lead you through all the testing.

The source code for both programs is included on the disk enclosed with this manual and both programs are discussed in detail, the USTPQB.BAS program in Appendix D and the USTPVB.VBP program in Appendix E.

UNIVERSAL SERIAL TEST PROGRAM (USTPQB)

The Universal Serial Test Programs are designed to thoroughly test every type of C/MRI serial node. The USTPQB version is written in QuickBASIC but many of the statements are similar to those used with many other languages. The program is heavily commented and makes abundant use of monitor printouts to lead you right through the testing sequence. For example, the program prints a list of all the required test conditions. It then asks you if you are testing an SMINI or an SUSIC and if it is an SUSIC, is it using 32- or 24-bit I/O cards. It asks you which PC COM port you are using and what baud rate is set on the card's DIP switch.

Also, the program asks if you are performing an Output Card Test or a Wraparound Card Test. Essentially, the same program works for all types of serial nodes. For the majority of C/MRI serial applications, this should be the only card test program you will need.

However, to support readers interested in writing their own card test programs, quite possibly in languages different from Basic, I have included a separate standalone program for performing an output card demonstration test, an input card demonstration test and a wraparound demonstration test. For the SMINI, these standalone programs are included in Chapter 7 and for the SUSIC they are the first three program examples covered in Chapter 12.

If you are using a language other than QuickBASIC or Visual Basic, feel free to use the source code for USTPQB or USTPVB programs, or the corresponding standalone programs, as baselines for converting to your selected language. Also, corresponding output card and wraparound card test programs for users of "C" and "Pascal" are included in the disk provided with versions of this manual prior to V3.0 and with the **Build Your Own Universal Computer Interface – Second Edition** book, published by McGraw-Hill. If you are using any of these alternative programs you will need to tailor the programs to fit the particular application you are testing.

The testing in this chapter assumes that you have a copy of QuickBASIC, QBasic, Power Basic or another compatible version of Basic installed on your computer that runs under DOS. Alternatively, if you are using USTPVB, this chapter assumes that you have a working version of Visual Basic. If this is not the case, then you need to obtain a copy of the appropriate program so that you can proceed. Otherwise, you will need to develop your own test programs, using a language of your choice.

Even if you are an extremely competent programmer and you want to do things on your own, I still recommend starting out using the "canned" USTPQB program, operating with some version of Basic under DOS for your initial hardware testing. Why? Because the biggest challenge encountered when first checking out an interfacing system is isolating whether an encountered problem is in the hardware or software. Using a well-proven universal testing program, like USTPQB, it can almost be guaranteed that any problem found is hardware related. However, if you are more acquainted with Visual Basic than with QuickBASIC, do not hesitate to start out using USTPVB.VBP in place of USTPQB.BAS.

Regarding Basic, I personally prefer QuickBASIC V4.5 mainly because it is readily available at little or no cost, it is easy to use, it works well for most C/MRI applications and it is the C/MRI language with which I am most familiar. Also, there are more C/MRI users making use of QuickBASIC and with its wide user base, it is easy to obtain programming assistance. The following assumes using the USTPQB.BAS program with QuickBASIC or its equivalent. If you are more interested in using the USTPVB.VBP program, consider checking out Appendix E if you would like more details about its operation.

CONDUCTING OUTPUT TEST

Card setup for SMINI application. With the computer connections complete, plug the test card onto the SMINI's output "Card 0" location and attach the clip lead to +5Vdc. If you are using the TEST32 card, simply leave the Port D portion of the test card hanging off the end of the SMINI card. Once we've successfully executed the USTP software for "Card 0" we'll move the test card to the SMINI's "Card 1" outputs and repeat the test. Alternatively, if you happen to have 2 test cards, the complete test can be run in a single step by attaching a separate test card to each of the SMINI's "output" cards. The test itself consists of driving the test card LEDs in a chase light pattern, starting with the A0 LED on "Card 0" and ending with the C7 LED on "Card 1".**[1]** This looping through all of the LEDs, turning on one at a time, automatically repeats itself until you terminate the program. Passing this test verifies that all 48 SMINI outputs are functioning correctly.

Card setup for SUSIC application. With the computer connections complete, turn off your +5Vdc power to the IOMBX and plug in a DOUT32 card set to address "0." Then plug a TEST32 card onto the DOUT32 card and attach the clip lead to +5Vdc. The test consists of driving the test card LEDs in a chase light pattern starting with the A0 LED on Port A and ending with the D7 LED on Port D. This looping of consecutively lighting one single LED at a time automatically repeats itself until you terminate the program. Passing this test verifies that all 32 outputs on the DOUT32 card are functioning correctly.

Bring up QuickBASIC (or its equivalent for your chosen programming language). How this is accomplished is computer/operating-system dependent. For example, the quickest way I have found with Windows 95/98 is to click on Start, Shutdown, Restart under DOS. Then after the DOS prompt, change directory to the location where you have stored QB.EXE and a copy of the Universal Serial Test Program (USTP). Then, after the DOS prompt, enter QB to bring up the QuickBASIC program. You may need to follow a different procedure - see Chapter 5 for alternative procedures and details.

Open USTPQB. With the QuickBASIC Menu page in place, click on Open and then select USTPQB.BAS. Once USTPQB is fully loaded, its source code will appear on the computer's monitor.

Run USTPQB. Pull down the Run menu and click on Start. Once activated, USTPQB will prompt your way through the complete Output Card Test. Simply answer each prompt correctly to initialize the test to fit your particular situation. Once initialization is complete, the program stops so that you can check that the SMINI's or SUSIC's green status LED shifts from its idle Mode 0 to Mode 1 (initialization is complete and receiving data without errors mode, with its blink rate of 1 second on and 1 second off).

Once you have checked out the blink rate, press any key to continue. At this point the USTPQB begins its real-time loop. Every time the program loops, the LED on the TEST32 card that's lit should move one position to the left. This pattern, with only one LED lit and moving from right to left, is typically referred to as a chase light pattern – the LED looks like its chasing something as it moves across the test card. At each position, compare the port and bit combination printed on the computer's monitor screen with the LED position lit on the Test Panel.

If you find the program executes too fast for you to accurately track the Test Panel LEDs, you can slow the process down by first stopping your program, by simultaneously pressing the Control and Pause/Break keys. Then restart USTPQB from scratch by clicking on Run and Start, but this time when asked, increase the value for slowing down the display.

---

[1] Note: The first SMINI production run cards labeled these "cards within a card" as 1, 2 and 3 versus the 0, 1 and 2 description provided in this manual. If you happen to have one of these early SMINI cards simply ignore the 1, 2 and 3 printing on the card and treat them as if they read 0, 1 and 2. See the section **Counting Cards/Nodes Using Number Zero** in Chapter 2 for the change in approach.

****Important Point****
Do not try to shortcut a restart of your program by clicking on Restart or by pressing the F5 function key. For result consistency, I recommend that you always perform a clean restart from scratch by pulling down the Run menu and then clicking on Start.

If the LED positions do not agree with the screen display, then you have some debugging to do. Once one port works correctly you are well on your way to total system success. As you encounter problems, do not hesitate to jump ahead and use the ***In Case of Difficulty*** section to provide assistance in helping you pass the Output Card Test.

Testing additional SMINI cards for SMINI applications. Once you have your first SMINI card testing properly then repeat the Output Card Test to test and debug all your other SMINI cards. This way you know that all your SMINI outputs are fully operational.

Testing additional DOUT32 cards for SUSIC applications. Once you have your first DOUT32 testing properly then repeat the Output Card Test to test and debug all your other DOUT32 cards. This way you know that all your output cards are fully operational.

This completes our testing of C/MRI outputs. Success obtained with the Output Card Test in the SMINI case proves that **most** of the SMINI is operational. Success with the SUSIC-node case proves that **most** of the SUSIC, IOMBX and all of the DOUT32 cards are operational. Also, for cases where the RS485 card is employed, a successful Output Card Test demonstrates that the output portion of that card is also functional as is the output portion of the cabling between the PC and the interface hardware.

In summary, a successful Output Card Test is a visual demonstration of the C/MRI's output capability but it does not yet verify that the inputs function correctly. However, the hardest part of the C/MRI project is behind you. Total interface success is just a small step ahead – testing that inputs can be sent to the computer.

If your Output Card Test worked the first time, you have earned the right to be elated! Most users who have done a careful assembly job find that they pass the above test the first time. If this is the case, you can simply proceed to the next section, ***Conducting Automated Wraparound Test.***

If your testing revealed a problem, take credit in that you discovered it. All this means is that there is some rechecking and debugging required. To help with debugging, temporarily jump ahead to the ***In Case of Difficulty*** section. Once you pass the Output Card Test, return to this point to begin wraparound testing.

CONDUCTING AUTOMATED WRAPAROUND TEST

Once you have successfully passed the Output Card Test it is time to test the inputs. We could test inputs by connecting a DIP switch to the card inputs and using a test program to read the switch settings and to display the results on the computer screen. Alternatively, without wiring a DIP switch, you could use clip leads to selectively ground input pins. However, with 24 inputs to test per SMINI and 32 per DIN32 card, it is much easier to use the computer to perform the whole test automatically. Automated testing and diagnostics is one of the many advantages of having a computer connected to your external hardware.

Because you have already successfully tested all the outputs, you can use them to test all the required inputs. In the computer industry this kind of I/O check is called a wraparound test. The computer's outputs are simply “wrapped around” and connected to its inputs. This wraparound capability is achieved using the wraparound test cable.

Very conveniently, we can use the same USTPQB (or USTPVB) software to perform the wraparound test. When prompted, you simply tell it you want to perform the Wraparound Card Test rather than the Output Card Test. When executing the wraparound test option, the test software writes a bit pattern to the output card and then reads the input card. It then compares what it reads with what it wrote, and if they don't agree it has found an I/O error and appropriately displays an error message. Since you have already proven the outputs are good, we can assume the fault is in the interface hardware’s input handling circuitry. If you get such errors, and the solution is not as obvious as might be indicated by a change in the green status light, consider jumping ahead and using the ***In Case of Difficulty*** section to provide assistance in helping pass the Wraparound Card Test.

To generate the outputs, the test software increments the value sent out on each port from 0 up to 255. This process checks every possible bit combination from 00000000 up to 11111111. Each time the program reads exactly what it wrote, it increments the bit pattern and repeats the write/read cycle. As the test for each port is completed, the test software moves on to repeat the process for each additional port. Once the last port is tested, the whole process is repeated ad infinitum until you stop the program by simultaneously pressing the control and pause/break keys.

After completing the full loop through all ports without errors, you know all your inputs are good as well as the outputs. To conduct the wraparound test, follow these instructions:

Card setup for SMINI application. Place a test card on the SMINI’s “Card 0” location and attach the clip lead to +5Vdc. If you are using the TEST32 card, simply leave the Port D portion of the test card hanging out off the end of the SMINI card. Now connect a wraparound test cable between “Card 1” and “Card 2” making sure that A0 connects to A0, B0 to B0, and so forth, all the way to C7 connecting to C7.

During the wraparound test for the SMINI, the USTPQB sends the same output data to both “Card 0” and “Card 1.” Therefore with the above setup, we can use the test card attached to “Card 0” to directly observe what is being sent out while “Card 1” is being used to wrap the same outputs back to the inputs.

Card setup for SUSIC application. Turn the power off to your IOMBX. It is not a firm requirement, but if you happen to have a second DOUT32 card it also is useful to set it also to the same Address 0 and to plug it into your IOMBX directly behind the first DOUT32 card. This provides two sets of outputs at the same address. By attaching a TEST32 card to the first DOUT32 card and a wraparound cable to the second DOUT32 card, we can observe the output from the test card simultaneously with the same data is being wrapped around from the second DOUT32 card.

To complete the setup, turn power off, plug in a DIN32 card set to Address 1. Attach the wraparound test cable between the DIN32 and the second (or first if you do not have a second) DOUT32 card. Make sure that A0 connects to A0, B0 to B0, and so forth all the way to D7 connecting to D7. If you are using the TEST32 card, connect its clip lead to +5Vdc. Now turn the power back on.

****Important Point****

You do not need to cycle the +5Vdc power when making connections to the I/O cards, including those built into the SMINI, but you should have the power turned off when plugging and unplugging cards into the IOMBX. Also, power should be turned off when making power supply connections to the SMINI and the IOMBX. Also, the power should be turned off whenever debugging requires IC removal and reinsertion.

Bring up QuickBASIC (or its equivalent for your chosen programming language). How this is accomplished is computer/operating-system dependent. For example, the quickest way I have found with Windows 95/98 is to click on Start, Shutdown, Restart under DOS. Then, after the DOS prompt, change directory to the location where you have stored QB.EXE and a copy of the Universal Serial Test Program (USTP). Then, after the DOS prompt, enter QB to bring up the QuickBASIC program. You may need to follow a different procedure - see Chapter 5 for alternate procedures.

Open USTPQB. With the QuickBASIC Menu page in place, click on Open and then select USTPQB.BAS. Once USTPQB is fully loaded its source code will appear on the computer's monitor.

Run USTPQB. Pull down the Run menu and click on Start. Once activated, USTPQB will prompt your way through the complete Wraparound Card Test. Simply answer each prompt correctly to initialize the test to fit your particular situation. I recommend you select the option to print every I/O line. To see what each line should look like you can peek ahead at Fig. 6-4. Each time the software finds a problem with the input not matching what was sent out, it prints an error message, as also illustrated in Fig. 6-4.

If you experience errors you can use the test card display to see exactly the bit pattern being sent to help locate which input bit is failing. To help solve such problems jump ahead to the ***In Case of Difficulty*** section.

****Important Point****

To get the Wraparound Card Test to function properly when input line filtering is employed, you most likely will need to include a time delay between sending outputs and reading inputs. This lets the inputs received, after filtering, reach the correct output values. The length of delay required depends upon the degree of filtering employed and the speed of your computer. Increasing the USTPQB (or USTPVB) INFILTER parameter stretches the time between sending outputs and reading inputs. Try a value of 5000 and then adjust up or down until you achieve satisfactory results.

Testing additional SMINI cards for SMINI applications. Once you have your first SMINI card testing properly for inputs, be sure to repeat the Wraparound Card Test to test and debug all your other SMINI cards. Thereby you know that all your SMINI inputs and outputs are fully operational.

Testing additional SUSIC cards for SUSIC applications. Once you have your first SUSIC card testing properly then repeat the Wraparound Card Test to test and debug all your other SUSIC cards. This way you know that all your SUSIC cards are fully operational to handle both inputs and outputs.

Testing additional DIN32 cards for SUSIC applications. Once you have your first DIN32 card testing properly then repeat the Wraparound Card Test to test and debug all your other DIN32 cards. You then know that all your input cards are fully operational.

This completes our wraparound testing of C/MRI cards. Passing both the Output Card Test and the Wraparound Card Test proves that your C/MRI hardware is fully functional. Knowing that you have good hardware prepares you for proceeding to Chapter 7 where we will look at ***Basic Programming Examples*** if you are using the SMINI, or to Chapter 12 where we will look at ***SUSIC Application Examples*** if you are using the SUSIC**.**

IN CASE OF DIFFICULTY

Most C/MRI users can skip this section because their C/MRI application will work the first time they turn it on. However, for users finding a problem, studying this section can provide a great deal of assistance.

If after receiving the keyboard inputs defining your particular computer and node configuration the Universal Serial Test Program (USTPQB) does not provide you with correct test results, you have what is known as an "interface communication problem." The USTPQB is sending outputs and they are not being correctly received by your interface hardware, or the USTPQB program is trying to read inputs from your hardware that are not being properly communicated. With either problem, the question is "How do I fix it?" The solution will probably be very simple once you find it but first you have to find it!

Remember, the most important requirement for problem solving is persistence, and the reward will be worth the effort. The following covers general things to look for in problem resolution, in addition to leading you through suggested debugging steps to solve specific problems. This way you can focus on solving specific problems, skipping those areas where you have the interface operational.

Program Operational Problems

To minimize the possibility of transmission errors, make sure you are using the lowest baud rate, 9600, for your initial testing. Then, if a PC overrun error is printed out on the PC monitor, the PIC16F877 most likely is feeding back inputs faster than the PC can handle them. Increasing the value of DL will slow down the 877's response.

If a MAXTRIES exceeded limit error appears, then the PC may be operating too fast for the PIC16F877. Try reducing the value of DL, which speeds up the SUSIC or SMINI. Also try increasing the value of MAXTRIES, which makes the PC wait for a longer period for receiving each input byte.

It is extremely important to make certain that you are running the USTP program strictly under DOS and not as a DOS program operating under Windows. Operating QuickBASIC programs under Windows will almost certainly introduce random communication errors.

> ****Important Point****
> When executing application programs, like USTPQB, with QuickBASIC, it is important to make certain that you are operating strictly under DOS without interference from Windows. If you encounter random I/O errors, then refer back to the sections in Chapters 2 and 5 dealing with Windows and its relationship to DOS and the use of QuickBASIC. Following the presented procedures helps to ensure operation without Windows interference. If you are operating with USTPVB, then these factors are not an issue.

If you still have problems when operating directly under DOS, it is time to make a closer inspection of your hardware.

Rechecking Card Assembly and System Connections

I repeatedly find that over 90 percent of all C/MRI problems can be found and solved by performing a careful visual inspection of each circuit board and the cable connection. That's the first step I take when anyone sends me C/MRI boards for checkout and repair.

I use a bright light and a magnifying glass to methodically search for assembly problems on each circuit card. I first study the component side, checking that each part number is correct, that each part is installed in the correct location with the correct orientation and that all IC leads are inserted correctly into their respective sockets. Every error found is corrected on the spot before proceeding with further inspection.

With the component side of each card checked and corrected, I then turn my focus to the solder side. If the solder flux is not completely removed, I clean the board, following the ***Cleaning and Inspection*** section in Chapter 1. Then, using my bright light and magnifying glass, I check every solder joint. I look at each joint from different angles to make sure the solder tent leaves no part of the hole exposed. I follow the procedures discussed in the ***PC Card Soldering*** section of Chapter 1, to check that solder is well attached to the lead as well as to the pad and that there are no solder bridges. Every error found is corrected on the spot before proceeding with further inspection.

Thorough inspections result in solving 90+ percent of all C/MRI problems and I do not proceed with any board testing until I have totally completed the visual inspection and correction cycle on both board sides. I strongly recommend that you implement this procedure!

Remember that most problems can be traced to faulty soldering, wrong or incorrectly inserted parts or cabling errors. Keep searching, and retesting until you find the last error, correct it, and achieve successful operation.

Performing a quality visual inspection is very much dependent upon **your state-of-mind**. Pretend that someone else assembled the boards. You do not trust that person to have done a good job and in fact your gut feeling is that there are definite assembly errors. Your whole mission now is to find every possible assembly error. You are now the senior person that will be checking this other person's work and you expect a large bonus for every error you find. In fact, you do receive a bonus because every error you find puts you one step closer to having a fully working interface. With this attitude and with careful inspection of both sides of every board, using a magnifying glass and a bright light, there is close to a 100 percent guarantee that you will find assembly errors.

Before you perform your second, or your third visual inspection make sure you have removed all traces of solder flux from every solder joint using a quality flux remover (see Chapter 1 ***Cleanup and Inspection*** section). Solder flux can very easily mask a poor solder connection.

Go back and reread every assembly step to make absolutely certain that the correct part is inserted with the correct orientation at every location. Make sure you have a quality solder tent around every component lead with none of the hole exposed. Review the section on ***PC Card Soldering*** in Chapter 1 and apply the techniques learned to every connection. Reheat each suspicious joint and apply a little more solder where it is required to form a perfect tent around each lead. Look from different angles at each solder joint for poor soldering or solder bridges. Look at each part for correct insertion with all pins in sockets, and so on.

Check that you have used the correct part number ICs in each card. Make certain that you followed the correct subsection for the serial standard you chose, RS232 or RS485. Recheck your serial computer connections per Fig. 4-9. If your computer's serial port is not set up to talk correctly, the SMINI or the SUSIC will not hear correctly. Note that the speed at which the yellow LED flickers is directly related to the baud rate.

If you are using the RS485 conversion card, make sure that +5Vdc and ground are connected and that all the communication connections are correct per Fig. 4-9. If your SMINI or SUSIC is assembled to also handle RS232 then bypass the RS485 card to eliminate it as a possible source of the problem. In doing this, make sure you temporarily remove the RS485 ICs and install the RS232 ICs.

Make sure you have the correct settings on the USIC address and baud rate DIP switches. Are all the switches installed with correct on/off orientation? If all switch settings are not correct for your C/MRI configuration, and if they do not correspond exactly to the address and baud rate you are using in your program and computer, then you will not get any response on the Test Card. For SUSIC applications make sure all the I/O cards are plugged onto the IOMBX with all pins inserted correctly and with the component side of the cards facing the correct direction.

Additional Debug Testing

If you have completed a determined-to-succeed visual inspection, and you are still not operational, it is time for more detailed hardware testing. To accomplish this, work your way through the following subtitles focusing on the areas that may be applicable to your problem. If working with the SMINI skip the SUSIC application steps and vice versa.

****General Note****
Always start with the simplest possible setup and get it working first: either an SMINI or an SUSIC with 1 output card. Use the RS232 connection – assuming your card is configured for RS232 and run the standard Universal Serial Test Program (USTPQB or USTPVB). Once you have the Output Card Test operational then, and only then, move on to getting the Wraparound Test operational with USTPQB or USTPVB.

Checking LEDs L1, L2 and L3 for SMINI applications. Temporarily remove U2. Use your VOM to check that Resistors R3-R5 measure close to the specified 330Ω ±5 percent. Then turn the power on and L1, L2 and L3 should be dark. Use your VOM to make sure the common connection going to each LED is at +5Vdc. Then connect a clip lead to ground and with a spare end of a resistor lead clipped in the other end touch Pin 8 in Socket U2 and L1 should light. Touch Pin 10 in U2 socket and L2 should light. Touch Pin 12 in U2 socket and L3 should light. Each LED that does not light is either inserted backwards or it is faulty. Replace each problem LED with a new one making certain that the + lead, usually the longer lead, is installed in the + hole with the square pad. Retest as necessary until all three LEDs pass this test. Then turn off the power and replace the removed IC.

Checking LEDs L1, L2 and L3 for SUSIC applications. Temporarily remove U2. Use your VOM to check that Resistors R3-R5 measure close to the specified 330Ω ±5 percent. Then turn the power on and L1, L2 and L3 should be dark. Use your VOM to make sure the common connection going to each LED is at +5Vdc. Then connect a clip lead to ground and with a spare end of a resistor lead clipped in the other end touch Pin 14 in Socket U2 and L1 should light. Touch Pin 16 in U2 socket and L2 should light. Touch Pin 12 in U2 socket and L3 should light. Each LED that does not light is either inserted backwards or faulty. Replace each problem LED with a new one, making certain that the + lead, usually the longer lead, is installed in the + hole with the square pad. Retest as necessary until all three LEDs pass this test. Then turn off the power and replace the removed IC.

Get green LED blinking in idle mode. When the PIC16F877 is powered up correctly the green status monitoring LED should begin blinking in its Mode 0 idle state, i.e., 1.8 seconds on and .2 seconds off. If not, first check that U1 is correctly receiving +5Vdc power and ground connections. You can verify this condition by setting your VOM to measure dc voltage and making the tests listed in Table 6-5.

Table 6-5. Correct power and ground connections to U1

√	IC	+ METER LEAD ON PIN No.	– METER LEAD ON PIN No.	VOLTAGE READING
	U1	11	12	+5Vdc
	U1	32	31	+5Vdc
	U1	1	31	+5Vdc

In performing the tests in Table 6-5, make sure you have your meter's negative lead connected to the indicated U1 pins and not simply connected to the card or power supply ground connection. If U1 is not receiving power and ground on each of the 3 lines indicated in Table 6-5 then trace the problem all the way back to the power supply. Look for a bad solder joint, a solder bridge or an improperly inserted or faulty part.

If U1 is receiving correct power and ground, make certain that you have the correct capacitor values for C15 and C16 if you are using the SMINI, and C9 and C10 if you are using the SUSIC. Values other than the specified 18pF can cause the crystal XL1 to not oscillate correctly and prevent U1 from operating. Be persistent in searching for such problems. For example, check part values and installation of R1, R2, D1 and C18 for the SMINI and C13 for the SUSIC, as well as associated soldering problems. If you still have a problem, try replacing U2.

If it is necessary to isolate the fault, you can remove all ICs except U1 and U2, because the others should have no effect on the green LED's indication. If this does not solve the problem, continue to look for bad solder joints or faulty connections in the area of U1, U2, R3 and L1. The cost of U1 includes program verification and full testing in a functional SUSIC, so it should be the very last thing to suspect in debugging your card.

Get yellow LED blinking when receiving data. The flickering of the yellow LED shows that data is reaching the SMINI or the SUSIC and is being passed through the U6 IC if you are using RS485 or through the U8 IC if you are using RS232. If it doesn't flicker at all, you have a basic interface problem in the setup and initialization of your computer's serial port, in the cabling between the port and the SMINI or SUSIC, or in the U2 and U6 or U8 circuit areas on the SMINI or SUSIC. Make certain if you are using RS232 that U8 is installed and that U6 and U7 are not installed and vice versa.

With the above correct, look for cable trouble, for instance serial connections that are not correct for your particular computer. Make sure all connections and wiring agree with Fig. 4-9a or b if you are using RS232 and Fig. 4-9c or d if using RS485. If you are using RS485, make sure you have not reversed the wiring of the two sets of transmission lines connecting the RS485 card with the SMINI or SUSIC. All wiring needs to be as defined in Fig. 4-9. If you are using RS232 you can try reversing the transmit and receive lines just to make sure your computer connector isn’t reversed.

If you are using the RS485 card make sure it is properly receiving +5Vdc and ground. To do so, set your VOM to read DC volts and touch the meter’s positive lead to the +5Vdc terminal on the RS485 card while holding the negative lead on the card’s ground screw. You should read +5Vdc.

The next most likely error is the wrong COM port number entered into the USTPQB or USTPVB program, i.e. the number entered does not reflect the actual COM port number to which the C/MRI hardware is connected. Restudy your computer manual, if you have one, to make sure you are addressing the COM port corresponding to the one that is connected to your C/MRI. If you do not have a manual, or if it is unclear, simply repeat the testing, each time trying a different COM port 1 through 4. Once you have found the one that blinks the yellow LED you have located the port to which you are connected.

Carefully touching the positive lead of your VOM meter to Pin 26 of U1, with the negative lead connected to ground, you should observe some activity when you have the correct COM port selected. If not, try replacing U6 or U8, depending upon whether you are using RS485 or RS232. If you see activity and you have already tested L2 as described above, then try replacing U2. Continue to look for bad solder joints or faulty connections until you have the problem solved.

At this point, baud rate settings are not a factor. The yellow LED simply shows that data is being sent over the transmission line and through the SMINI's or SUSIC’s I/O-unique circuitry. Because L2 is located before the PIC16F877, satisfactory testing of this step is more a function of having a correct setup at the computer rather than a test of the SMINI or SUSIC.

Get green LED blinking without errors after initialization. If the SMINI or SUSIC accepts the initialization from the PC as valid, the PIC16F877 changes the green LED's blink rate from Mode 0 to Mode 1 per Table 6-3. This is normal operation.

However, if the PIC16F877 detects some type of error occurring during initialization, it changes the LED Mode from 0 to one of the error modes, i.e. Modes 2 through 6. Typically this occurs when bad initialization data is sent from the PC to the 877. Using the indicated blinking pattern, Table 6-3 and the discussion provided in the section ***Using the Status Monitoring LEDs*** should put you on the right track to solving the problem.

IC power tests for SMINI and SUSIC. A vital requirement for a functional interface is that correct power is reaching each IC on the SMINI and SUSIC. You can verify this condition by setting your VOM to measure dc voltage and making the tests listed in Table 6-6 for SMINI and Table 6-7 for SUSIC.

Table 6-6. SMINI IC power tests

√	IC	+ METER LEAD ON PIN No.	– METER LEAD ON PIN No.	VOLTAGE READING
	U1	11	12	+5Vdc
	U1	32	31	+5Vdc
	U1	1	31	+5Vdc*
	U2	14	7	+5Vdc
	U3	20	10	+5Vdc
	U4	20	10	+5Vdc
	U5	20	10	+5Vdc
	U6	8	5	+5Vdc
	U7	8	5	+5Vdc
	U8	16	15	+5Vdc
	U8	2	15	+10Vdc
	U8	15	6	+10Vdc
	U9	20	10	+5vDC
	U10	20	10	+5vDC
	U11	20	10	+5vDC
	U12	20	10	+5vDC
	U13	20	10	+5vDC
	U14	20	10	+5vDC

No application uses all ICs. Should read indicated voltage on all that are installed
*Reading may be slightly less than other +5Vdc readings

Table 6-7. SUSIC IC power tests

√	IC	+ METER LEAD ON PIN No.	– METER LEAD ON PIN No.	VOLTAGE READING
	U1	11	12	+5Vdc
	U1	32	31	+5Vdc
	U1	1	31	+5Vdc*
	U2	20	10	+5Vdc
	U3	20	10	+5Vdc
	U4	20	10	+5Vdc
	U5	20	10	+5Vdc
	U6	8	5	+5Vdc
	U7	8	5	+5Vdc
	U8	16	15	+5Vdc
	U8	2	15	+10Vdc
	U8	15	6	+10Vdc

No application uses all ICs. Should read indicated voltage on all that are installed
*Reading may be slightly less than other +5Vdc readings

Check off each line where you obtain a correct reading. If you find an IC not getting power, work back along the circuit path to locate a part insertion problem or an open circuit. A low voltage, less than 4.8Vdc, most likely means an inadequate power supply or a short circuit on your card caused by a solder bridge or a part installed in reverse. Turn off the power and re-examine the card to find and correct the error.

If you have a short and only if necessary, remove ICs one at a time to isolate the problem. If you do trace a problem to an IC, replace it with a new part. Also make sure that polarity sensitive capacitors are correctly installed. Make sure you discard any part determined to be faulty. It is handy to keep at least one spare of each IC type used in your C/MRI for debugging and possible replacement.

Also feel each IC for a high temperature. If an IC is hot to the point where you cannot hold your finger on the part, the fault is likely to be:

1. installed backwards
2. the wrong part
3. shorted

It is possible that an overheated part may still function but the best recourse is to simply throw away any part that has been overheated.

DIP switch testing for SMINI and SUSIC. As discussed in Chapter 1, the labeling of the DIP switches and their orientation can be a problem. If you have used the specific DIP switch source and part number specified in the parts list and installed them as described, then you should have no trouble. However, to verify correct orientation as well as correct operation, use your VOM to perform the measurements listed in Table 6-8.

Table 6-8. SMINI and SUSIC DIP switch tests

√	IC	IC Pin No.	DIP Switch	Switch Segment
	U1	2	SW1	A0
	U1	3	SW1	A1
	U1	4	SW1	A2
	U1	5	SW1	A3
	U1	6	SW1	A4
	U1	7	SW1	A5
	U1	23	SW1	A6
	U1	15	SW2	19200
	U1	16	SW2	28800
	U1	17	SW2	57600
	U1	18	SW2	115200

With switch segment set to “ON”, up position on card, IC pin should read 0Vdc
With switch segment set to “OFF”, down position on card, IC should read +5Vdc

Remember from Chapter 1, when performing DIP switch testing, do not pay any attention to the labeling on the DIP switch itself, follow only what is printed on the circuit board and the instructions printed in this manual.

If you find a problem it will most likely be caused by RN1 or RN2 being installed backwards or faulty, +5Vdc not getting to Pin 1 on RN1 or RN2, a faulty DIP switch or a soldering problem in the area of RN1, SW1, RN2, SW2 and the associated pins on U1.

Most but not all output test LEDs function properly, for SMINI applications. If you have reached this point in your testing and debugging you are close to final success. First make sure you have thoroughly tested your Test Card as defined in the earlier section titled ***Test32 Card Testing.*** Then with a verified Test Card you can make sure that all 48 of the transistor driver circuits set up for standard current-sinking are working correctly by following the steps:

1. Make sure that jumpers J1-J6 are in their long position on SMINI.
2. With power off, remove ICs U9-U14
3. Turn power on and all 48 test panel LEDs should be on.
4. Temporarily grounding each of Pins 12 through 19 on the U9 through U14 sockets should individually turn off the corresponding LED.

For each LED that fails to function in this manner you either have a soldering problem, RN6 through RN11faulty, RN12 through RN17 faulty or installed backwards, +5Vdc not getting to the Pin 1 connections on RN12 through RN17 or a failed transistor. If you cannot locate a soldering problem and the transistor network resistances check out OK then replace the transistor. Repeat the test to make sure that each LED works correctly.

Most but not all output test LEDs function properly, for SUSIC applications. If you have reached this point in your testing and debugging you are close to final success. First make sure you have thoroughly tested your Test Card as defined in the earlier section titled ***Test32 Card Testing.*** Then, if you have other DOUT32 cards substitute them into your test to help determine if the problem is associated with the SUSIC or the DOUT32 or possibly even with the IOMBX. If you find a DOUT32 card that functions correctly then use it as a reference to substitute parts with those that have problems and to seek out and correct other differences. Also try using different card slots to help locate possible faulty soldering problems with your IOMBX.

A quick and easy check to make sure that all 32 of the transistor driver circuits on a DOUT32 card, set up for standard current-sinking are working correctly is to:

1. Make sure that jumpers J1-J4 are in their long position on DOUT32.
2. With power off remove ICs U3-U6
3. Turn power on and all 32 test panel LEDs should be on.
4. Temporarily grounding each of Pins 12 through 19 on the U3 through U6 sockets should individually turn off the corresponding LED.

For each LED that fails to function in this manner you either have a soldering problem, RN2 through RN5 faulty, RN6 through RN9 faulty or installed backwards, +5Vdc not getting to the Pin 1 connections on RN6 through RN9 or a failed transistor. If you cannot locate a soldering problem and the transistor network resistances check out OK then replace the transistor. Repeat the test to make sure that each LED works correctly.

Certain output ports function properly but others do not, for SMINI applications. Assuming you have proved that all the transistor driver circuits are functioning correctly then the problem is most likely associated with faulty soldering associated with U1 or U9-U14 or bad ICs. If you cannot locate soldering problems then temporarily move the output latch IC, U9-U14 (a 74HCT573) from a port that works to each one that doesn't. If the offending port now works replace the bad 74HCT573. Keep comparing a port that works to one that doesn't to seek out and correct differences.

Certain output ports function properly but others do not, for SUSIC applications. If you have other DOUT32 cards, substitute them into your test to help determine if the problem is associated with the SUSIC or the DOUT32 or possibly even with the IOMBX. If you find a DOUT32 card that functions correctly then use it as a reference to substitute parts with those that have problems, seek out and correct other differences. Also try using different card slots to help locate possible faulty soldering problems with your IOMBX.

Assuming you have proven that all the transistor driver circuits are functioning correctly, then the problem is most likely associated with faulty soldering associated with U1 or U9-U14 or bad ICs. If you cannot locate soldering problems then temporarily move the output latch IC, U9-U14 (a 74HCT573) from a port that works to each one that doesn't. If the offending port now works replace the bad 74HCT573. Keep comparing a port that works to one that doesn't until you discover the difference and then fix it so they are the same.

Every Port Has the Same Error, for SMINI applications. If you have the same error in every port then the problem is most likely in the data bus traces that run from U1 to each of the latches U9-U14 and the input buffers U3-U5. Look for bad soldering and especially for a solder bridge between IC pins associated with this bus. Check out the bus connections to associated pins on U3-U5 and U9-U14. If no problem is found, temporarily pull U3-U5 to see if the problem corrects itself. If so replace the offending IC(s).

Every Port Has the Same Error, for SUSIC applications. If you have the same error in every port and switching DOUT32 cards does not correct the problem, then the problem is most likely in the SUSIC. Look for bad soldering in the area of U3, U4 or U5 and their connections to U1.

If this does not solve the problem you may have the same problem on each of your DOUT32 cards. Check the DOUT32 cards to make sure that U7 is the correct part. Check the data bus traces associated with U7 and their parallel paths to U3-U6. If you cannot find a problem, try replacing U7. Look for bad soldering and especially for a solder bridge between IC pins associated with this bus.

System works OK with RS232 but not with RS485. Observe performance of the LEDs on the RS485 card to help identify and correct problem. Also check the wiring to and from the RS485 card to make sure it agrees with Fig. 4-9c or d. Check for hot ICs, especially the U1 RS485 driver IC. If an IC is hot, check that the part is the correct part and that it is installed correctly. The suggested next step is to use your VOM to make the power tests listed in Table 6-9.

Table 6-9. RS485 conversion card IC power tests

√	IC	+ METER LEAD ON PIN No.	– METER LEAD ON PIN No.	VOLTAGE READING
	U1	8	5	+5Vdc
	U2	8	5	+5Vdc
	U3	14	7	+5Vdc
	U4	16	15	+5Vdc
	U4	2	15	+10Vdc
	U4	15	6	+10Vdc

If you do not find correct voltages, check for soldering faults, a wrong part or a part installed backwards and follow the circuit traces back to find the fault.

If you have proper power to the IC pins but the LEDs still do not function properly, it is time to check the LEDs. With power applied attach one end of a clip lead to the ground screw, and clip a piece of bare, solid wire in the other end as a test probe. Carefully touch the probe to Pin 12 of U3 and L1 should light. Touch Pin 6 of U3 and L2 should light. If not, then you can check soldering around the LEDs and U3 but most likely the problem is a LED in backwards or a faulty LED. If after successfully checking the power and LEDs you still have a problem, then your best bet is to replace ICs to see if you can find one that is faulty. If so, be sure to discard the faulty IC.

Outputs all work OK but inputs do not, for SMINI applications. If none of the inputs work and you have other SMINI cards substitute them into your test to help determine if the problem is associated with the SMINI or with the wiring back to the PC or possibly in the configuration of the PC serial port. If you find an SMINI that works use it as a reference to substitute parts with those that have problems and to seek out and correct other differences. Be careful with part swapping. For example, if you find an IC is hot, a short is likely to be present. Try to find it first so that additional parts are not destroyed. Typically a hot IC is the result of a part inserted backwards or an incorrect part.

Try replacing U8 if you are using RS232 or U7 if you are using RS485. Also check for incorrect or faulty wiring between the SMINI and your PC. If the problem is only with certain ports, then substitute the input buffer IC (U3, U4 or U5) from a port that works to one that doesn't. Replace any faulty parts you find with a new part.

To check the operation of each input line, remove all input connections and use your VOM to measure the DC voltage at each input pin. They should all measure +5Vdc. If you find some that don't then, with the power off, temporarily remove U3 through U5 and repeat the test. If you still don't get +5Vdc on all input pins look for RN3 through RN5 inserted backwards or faulty, or the +5Vdc not getting to the Pin 1 connection on RN3 through RN5 and possible soldering problems. If the +5Vdc is present without U3 through U5 installed but not when installed, then replace each of the ICs associated with the lines that are not working.

Outputs all work OK but inputs do not, for SUSIC applications. If none of the inputs work and you have other DIN32 cards substitute them into your test to help determine if the problem is associated with the DIN32 card, the input portion of your SUSIC, the wiring back to the PC or possibly in the configuration of your PC serial port. If you find a DIN32 card that functions correctly, then use it as a reference to substitute parts with those that have problems and to seek out and correct other differences.

If trying multiple DIN32 cards does not solve the problem, then try replacing SUSIC IC U8 if you are using RS232 or U7 if you are using RS485. Also check for incorrect or faulty wiring between the SUSIC and your PC. If the problem is with only certain ports then substitute the input buffer IC on the DIN32 cards, namely U3, U4, U5 and U6 from a port that works to one that doesn't. Replace any faulty parts you find with a new part.

To check the operation of each DIN32 input line, remove all input connections and use your VOM to measure the DC voltage at each input pin. They should all measure +5Vdc. If you find some that don't then, with the power off, temporarily remove U3 through U6 and repeat the test. If you still don't get +5Vdc on all input pins, look for RN2 through RN5 to be backwards or faulty, or the +5Vdc is not getting to the Pin 1 connection on RN2 through RN5 and possible soldering problems. If the +5Vdc is present without U3 through U6 installed but not when installed then replace each of the ICs associated with the lines that aren't working.

DIP switch testing for DOUT32 and DIN32 cards, for SUSIC applications. As discussed in Chapter 1, the labeling of DIP switches and their orientation can be a problem. If you've used the specific DIP switch source and part number I have indicated in the parts list and installed them per directions then you should be all right. However, to verify both correct orientation and operation, use your VOM to perform the measurements listed in Table 6-10.

Remember from Chapter 1, when performing DIP switch testing, do not pay any attention to the labeling on the DIP switch itself, follow only what is printed on the circuit board along and the instructions printed in this manual. Of special note is that the correct readings in Table 6-10 for the I/O cards are exactly opposite those obtained with the SMINI and SUSIC as denoted in Table 6-8.

Table 6-10. DOUT32 and DIN32 DIP switch tests

√	IC	IC Pin No.	DIP Switch	Switch Segment
	U2	3	SW1	A2
	U2	5	SW1	A3
	U2	7	SW1	A4
	U2	9	SW1	A5
	U2	12	SW1	A6
	U2	14	SW1	A7

With switch segment set to “ON”, up position on card, IC pin should read +5Vdc
With switch segment set to “OFF”, down position on card, IC should read 0Vdc

If you find a problem it is most likely caused by RN1 being installed backwards or faulty, +5Vdc not getting to Pin 1 of RN1, a faulty DIP switch or a soldering problem in the area of RN1, SW1 and the associated pins on U2.

IC power tests for DOUT32 and DIN32, for SUSIC applications. A vital requirement to have a functional interface with the SUSIC is that correct power is reaching each IC on the DOUT32 and DIN32 cards. You can verify this condition by setting your VOM to measure dc voltage and making the tests listed in Table 6-11. Everything is the same for both cards except that the U7 and U8 readings are not present on the DIN32 card.

Table 6-11. DOUT32 and DIN32 IC power tests

√	IC	+ METER LEAD ON PIN No.	– METER LEAD ON PIN No.	VOLTAGE READING
	U1	16	8	+5Vdc
	U2	20	10	+5Vdc
	U3	20	10	+5Vdc
	U4	20	10	+5Vdc
	U5	20	10	+5Vdc
	U6	20	10	+5Vdc
	U7	20	10	+5Vdc
	U8	14	7	+5Vdc

Check off each line where you obtain a correct reading. If you find an IC not getting power, work back along the circuit path to locate a part insertion problem or an open circuit. A low voltage, less than 4.8Vdc, most likely means an inadequate power supply or a short circuit on your card caused by a solder bridge or a part installed backwards. Turn off the power and re-examine the card to find and correct the error.

If you have a short and if necessary, remove ICs one at a time to isolate the problem. If you do trace a problem to an IC, replace it with a new part. Also make sure that polarity sensitive capacitors are correctly installed. Make sure you discard any part determined to be faulty. It is handy to keep at least one spare of each IC type used in your C/MRI for debugging and possible replacement.

Complete IC Replacement. If after doing all the above you still cannot find your problems (and you are confident that your soldering is good, that you have correct parts installed at all locations with the correct orientation) then consider a complete replacement of all of the ICs. You may just have a faulty one that has eluded detection. Leave U1 on the SMINI and SUSIC till last as it was thoroughly tested before it was shipped to you. You have not lost much with this action because it is good to have a complete set of ICs on hand as spares. However, always make sure you discard every IC you determine to be faulty so there is zero chance it becomes mixed in with your good parts.

ADDITIONAL SUPPORT

The C/MRI User's Group at http://groups.yahoo.com/group/CMRI_Users/ is a great source of support to answer questions concerning the C/MRI and to help get a system operational. Also, the members of this group are a great source for applications support.

Also, you can try to get help from a computer expert in your area. In many computer clubs you will find hobbyists just looking for tough problems to solve. The technician at your computer service center may be able to help you on the side.

If you live near a college or school teaching electronics, check out the digital electronics classes. A student may be able to help you and earn course credit for doing it, plus a payment from you for helping out. An experienced person with the aid of an oscilloscope and/or a logic analyzer should be able to find almost any interface problem with minimum effort.

Another idea is to get a friend, preferably one with some electronics background, to go through all the checks with you. Often someone else will find something wrong where the person who made the error passes right over it. As you perform the checks, have your friend challenge you with questions like, "How do you know that is the correct part?" and, "Are all the wires in that connector making contact?" As you try to explain what you're doing at each step you will very likely find your problem. In industry we call this a "design walk-through."

Also, Marc Robertson at SLIQ Electronics, and Bruce Chubb at JLC Enterprises do provide quality, over the telephone or by email, support without charge, provided that your cards were purchased from either SLIQ Electronics or JLC Enterprises. However, extensive support beyond a reasonable time limit does require payment of an hourly fee. For additional support, you can send in your cards for testing and repair. A nominal hourly rate is charged plus the cost of parts and return shipping. Both Marc and Bruce are extremely knowledgeable about the system and typically within an hour or two they can have your system in tip-top shape. However, always make contact by telephone or email for instructions before sending your cards in for testing and repair.

Another tremendous support center is the C/MRI User's Group located at:

http://groups.yahoo.com/group/CMRI_Users

It is likely that other users have experienced the same problem you may be having and can provide quick and very effective support.