Notes on Using Busprobe


Notes on Using Busprobe.doc

What is Busprobe?	Busprobe is a software application that ships on the hard drive of GSM machines. It can be used to bypass the machine controller and access I/O boards on the MMIT bus: to read machine inputs and write machine outputs to and from these I/O boards.

When to use Busprobe? When I/O boards fail, the failure can cause an initialization error during platform power-up. If the platform does not completely initialize, you will be unable to use the Discrete I/O feature in USOS to check the MMIT I/O boards. Under these conditions, you can check the I/O boards using Busprobe. (You can also capture initialization messages using Logicomm.)

WARNING!	Improper selection of address or output bits can cause unexpected machine actions. Using Busprobe is not really difficult, but there are several steps in which it is possible to make a simple mistake.

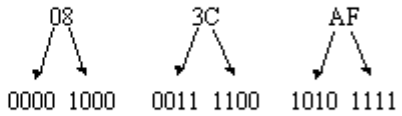
Materials	* GSM (platform) machine * FAW sheets from the machine's Maintenance Support Documentation

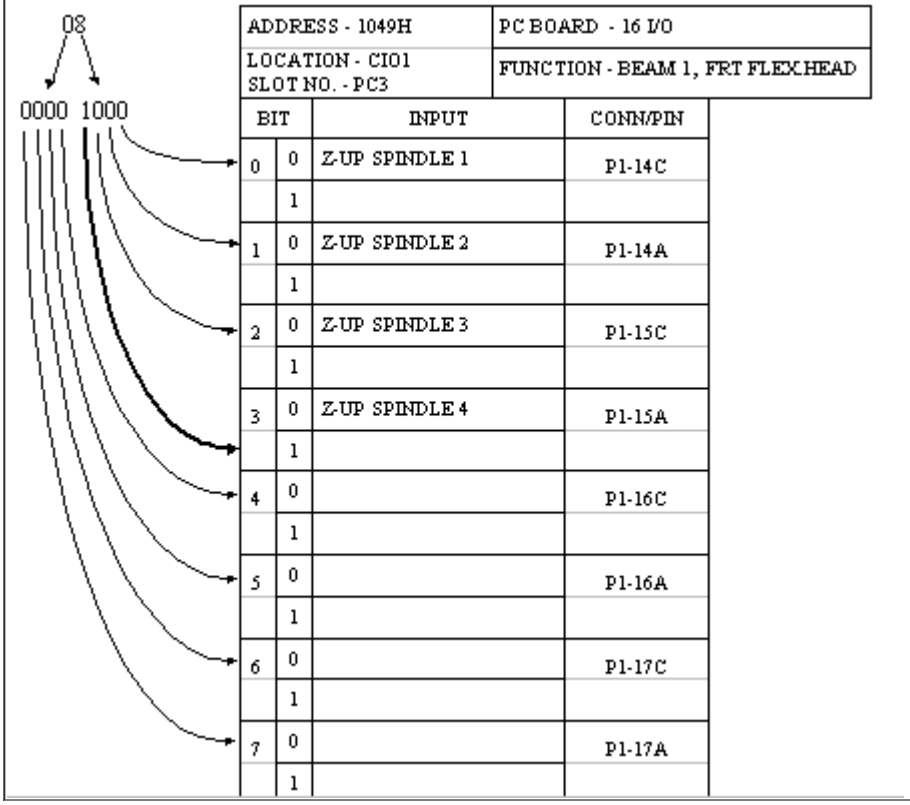
To Start Busprobe

Do this:	1. If the machine controller (Force bd.) in the VME has a R/LRES toggle switch (found on earlier rev. Force boards), toggle the R/LRES switch to the up position. If the machine controller does not have a R/LRES toggle switch, continue with step 2. Note: The Force board is located in slot 5 on newer GSMs, and in slot 7 of older GSMs.
	2. Open an OS/2 window by selecting the OS/2 Window icon from the main USOS window. If you are running older USOS software and do not have an OS/2 Window icon: * press [Ctrl]+[Esc] to open the Window List * Double-click on the OS/2 Window application in the Window List
	3. At the C: prompt, type: **cd\usos\epconnec** and press [Enter].
	4. At the C:\USOS\EPCONNEC prompt, type: **busprobe** and press [Enter].
	5. From the Busprobe title bar, select **VME** and select either **Read Data** (to check inputs) or **Write Data** (to check outputs).
	6. In the Bus Access Parameters window, make the following selections and then select [OK].

	* Supervisor Data - **A16** * Data Width - **D08(EO)** * Byte Order - **Motorola**
	7. If the Num LED on the keyboard is not lit, press [Func]+[Scroll Caps/Num] to enable the keyboard's number keys.
	8. In the machine's Maintenance Support Documentation locate the FAW Electrical Diagram for the input or output function that you wish to check. Some functions have similar names and may appear to be repeated in the FAW diagrams -- but they are *not* repeated. Check the Function box of the FAW sheets to ensure that you have the FAW sheet for the correct Beam, Head, etc.
	9. Select the Addr: box in the Read or Write window, and enter the first 4 digits of the address from the appropriate FAW sheet. Example: If the address on the correct FAW sheet is "1049H" enter only "1049".

Using Busprobe to Check Inputs (Read Data)

Do this:	Note: You must have selected the Read Data item from the VME title bar option, and entered an address in the Addr box. **Do not change the value in the Len field.**
	1. Press [Enter] or use the trackball to select the Read button.
	2. The address and a 2-character alphanumeric code is displayed in the Bus Probe window. Convert each of the 2 characters to its 4-digit binary equivalent. See the conversion table and example below. **Conversion Table** **Hex Binary Hex Binary** 0 0000 8 1000 1 0001 9 1001 2 0010 A 1010 3 0011 B 1011 4 0100 C 1100 5 0101 D 1101 6 0110 E 1110 7 0111 F 1111 **3 Examples of Hex-to-Binary Conversion** 
	3. Compare the 8-digit binary number to the INPUTs listed in the FAW sheet. Note that the right-most digit of the 8 digits relates to BIT 0 in the FAW sheet; the left-most digit relates to BIT 7 in the FAW sheet. The diagram on the next page shows how the binary numbers correspond with the INPUT bits in the FAW sheet. **The INPUT side of a FAW Sheet**



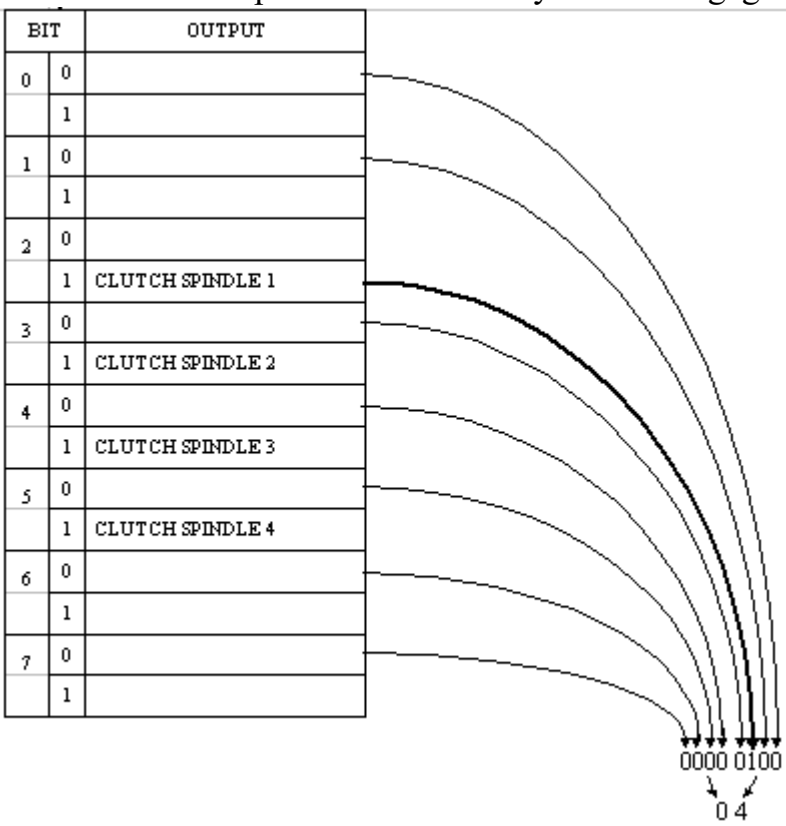
ADDRESS - 1049H	PC BOARD - 16 I/O
LOCATION - CIO1 SLOT NO. - PC3	FUNCTION - BEAM 1, FRT FLEXHEAD

BIT		INPUT	CONN/PIN
0	0	Z-UP SPINDLE 1	P1-14C
	1		
1	0	Z-UP SPINDLE 2	P1-14A
	1		
2	0	Z-UP SPINDLE 3	P1-15C
	1		
3	0	Z-UP SPINDLE 4	P1-15A
	1		
4	0		P1-16C
	1		
5	0		P1-16A
	1		
6	0		P1-17C
	1		
7	0		P1-17A
	1		

	The Busprobe input data in the above example indicates that the Z-up switch on spindle 4 (on the front flex head on beam 1) is *not* made.

Using Busprobe to Check Outputs (Write Data)

To Write Data	Notes: You must have selected the Write Data item from the VME title bar option, and entered an address in the **Addr** box. Do not change the value in the **Len** field.
	1. Determine the 8-digit binary code from the output bits listed in the FAW sheet. Then convert these binary numbers to hexadecimal. An example is given

	below. This example shows the binary code to engage the clutch on spindle 1. 
	2. Convert the binary code to "hex." Refer to the Conversion Table and the example above.
	3. Select the Data box in the *Bus Probe - Write* window, and type the 2-character hex code, and select the Write button (or press [Enter]). The output function(s) selected should now take place on the GSM.

Using Busprobe to Check PCB's with Diagnostic Bytes

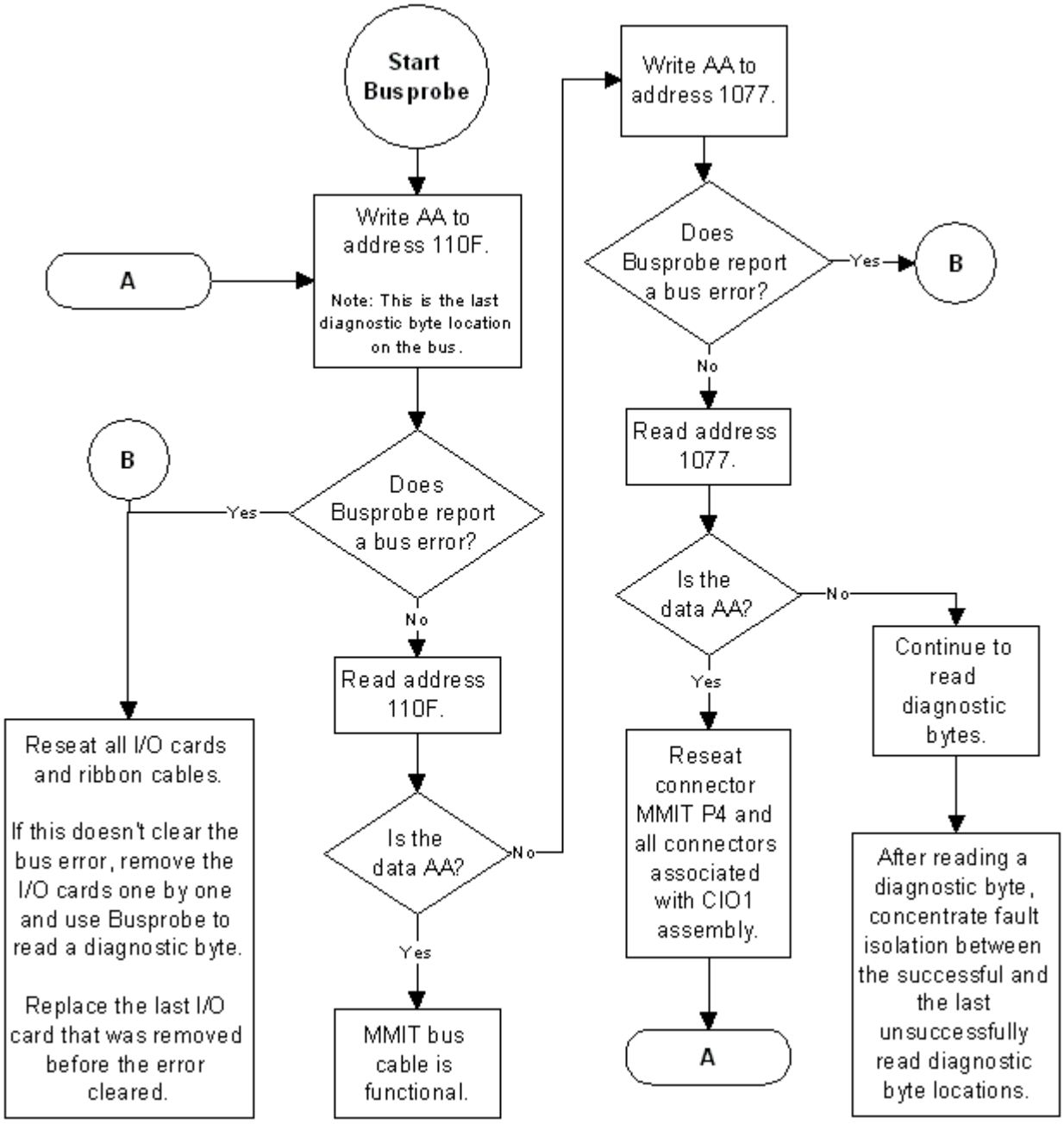
What is a diagnostic byte?	Some PC boards have a built-in "diagnostic byte" function. A technician can use this function to perform a health check on the I/O bus and the PC board in question.
How does the diagnostic byte work?	A PC board with a diagnostic byte function has a diagnostic byte address in addition to its normal address(es). An output byte is sent to this diagnostic byte address. If all is working well, the PC board returns a diagnostic input byte that is the same as the output byte that was sent.
How do I check diagnostic byte addresses	The table below shows the PC boards on the GSM that have a diagnostic byte function.

on the GSM?	<table><tr><th>Location</th><th>Slot Number</th><th>Hex Address</th></tr><tr><td>I/O Chassis Assy</td><td>PC10</td><td>100F</td></tr><tr><td>I/O Chassis Assy</td><td>PC11</td><td>101F</td></tr><tr><td>Feeder Interface</td><td>FD1</td><td>1077</td></tr><tr><td>Feeder Interface</td><td>FD2</td><td>107F</td></tr><tr><td>Feeder Interface</td><td>FD3</td><td>1087</td></tr><tr><td>Feeder Interface</td><td>FD4</td><td>108F</td></tr><tr><td>HP Servo Chassis</td><td>PC4</td><td>110F</td></tr></table>
	If the PC board in question has a diagnostic byte function, then: 1. Use the *Write Data* function in BUSPROBE to send an output to the diagnostic byte address of the PC board. 2. Then use the *Read Data* function in BUSPROBE to read the diagnostic byte address of the board.

How do I interpret the results?	If the data you read is the data you wrote, then the bus and the board look OK. If you are unable to read the same byte as you wrote: * the I/O PC board is not functioning, or * there is a break in the bus between the MMIT to PCB interface card and the I/O PCB under test. If Busprobe returns a bus error: * an I/O PC board with this address is not present on the bus, or * the I/O PC board at this address is not functioning.

Using Busprobe to Isolate GSM Bus Errors

Directions: Use the following flow chart to isolate and correct a fault in the MMIT bus.



To Exit from Busprobe

Do this:	1. Close the *Write* and/or *Read* windows by double-clicking on the box to the left of the title bar(s). 2. To close the Bus Probe main window, double-click on the box to the left of the title bar. You will be prompted: "Save transactions to log file:". Select the [No] button. 3. Minimize the OS/2 window by selecting the minimize button in the upper right corner.

	4. If the machine controller (Force bd.) in the VME has a R/LRES toggle switch (older GSMs), toggle the R/LRES switch to the down position (this is the normal position for machine operation). If the machine controller does not have a R/LRES toggle switch, continue with step 5. 5. Power cycle the machine.

01 Oct 98 CB

Click on the arrow on the upper left corner to return to the previous page (⇦).