



Sender Board Theory of Operation

Document Number: TR-2-2015

Document Version: 1.0

Sender Board Theory of Operation

Table of Contents

1	Introduction	3
2	Board Version Notes	4
2.1	Design Prototype.....	4
2.2	Version 1.....	4
2.3	Version 2.....	4
2.4	Version 3.....	4
3	Circuit Description	5
3.1	Block Diagram.....	5
3.2	Bus Interface.....	5
3.2.1	Version 1 and Version 2.....	5
3.2.2	Version 3.....	5
3.3	Select & Parallel Port.....	6
3.4	Clock Generators.....	7
3.5	DCC Signal Generator.....	8
3.6	External Interface.....	9
3.6.1	RS-422 Outputs and Inputs.....	9
3.6.2	6 TTL Level Outputs.....	9
3.6.3	8 TTL Signal Inputs (Version 2 & 3 Only).....	9

Associated Documents

- [Sender V2 System Document Folder](https://app.box.com/s/pfwr5yevja99bk1fpr1so5lli8n5mgbn)
 - Sender V2 Schematic.pdf
- [Sender V3 System Document Folder](https://app.box.com/s/vkkzel32wlmhs3nzktazhwrp6u5ibkke)
 - TN9.1.3 Sender V3 Getting Started Guide.pdf
 - Sender V3 Schematic Send104E.pdf

Sender Board Theory of Operation

1 Introduction

The NMRA conformance test Sender Boards provides the necessary low level DCC signal to drive the conformance testing booster. The resulting signal is then used to test a DCC receiver. The sender board has the following major features:

1. The board produces an RS-422 compliant balanced DCC signal output. The DCC signal output is suitable for driving the booster.
2. The board can generate DCC 1 and 0 signals with a resolution of 1 microsecond. The low and high time of the 0 pulse can be set independently to allow for *stretched* 0 pulses. The 1 pulse is always sent with a 50% duty cycle. The basic clock generator is crystal controlled to provide a highly accurate DCC signal.
3. The board provides an RS-422 compliant scope trigger signal that is controlled by software and provides a hardware generated synchronization pulse at the beginning of a selected portion of the DCC signal. This scope trigger may be used to synchronize an oscilloscope with the DCC signal.
4. The board provides 4 RS-422 compliant balanced inputs which can be directly read by the software. These inputs can be used to send data on motor direction, lamp status, etc., to the board.
5. The version 2 and version 3 board provide an additional 8 TTL compatible inputs.
6. Different versions of the board support different interfaces to the host IBM compatible computer and the DCC test system. The board uses an 8 bit interface. The data is transferred using I/O port **IN** and **OUT** instructions. The specific interfaces supported by each board version are shown below:

Version 1: The bus interface supports the 8 bit IBM ISA bus and ISA form factor.
 The DCC test system signals use a board-mounted DB-25S connector.
 A data request interrupt may be selected by jumper JP301.

Version 2: The bus interface supports the 8 bit ISA bus and ISA form factor.
 The board also supports the PC/104 interface.
 The DCC test system signals use a board mounted DB-25S connector and a board
 mounted DB-9S connector.
 A data request interrupt may be selected by jumper JP301.

Version 3: The bus interface supports the PC/104 interface and the PC/104 form factor.
 The DCC test system signals use a board mounted 2x13 pin array that may be connected
 to a DB-25S cable using a standard ribbon cable. The additional 8 input signals used a
 board mounted 2x5 pin array.
 A data request interrupt is not supported.

7. The board provides special self test logic to provide a software controlled clock and read back signals from several places along the signal path. This allows the software to conduct a thorough self test prior to operation. This additional hardware also helps isolate hardware problems.

Sender Board Theory of Operation

2 Board Version Notes

2.1 Design Prototype

The prototype of the sender board is built on the PDS-601 Breadboard from JDR Microdevices, 2233 Samaritan Drive, San Jose, Ca., 95124. The board consists of dedicated PC bus decode and buffer logic and a section of breadboard. The board works well for prototyping but has caused some problems with intermittent connections. A much better solution would be to build a printed circuit board for this circuit.

The board is constructed using standard 74LS series parts. It would be possible to replace several of these parts with one or two programmable devices (PALs, etc.) to reduce the part count. This would require access to a PAL programmer.

2.2 Version 1

Version 1 was laid out using an 8 bit ISA interface and ISA form factor.

The board is constructed using standard 74LS series parts except for the 20V8 PAL U302.

2.3 Version 2

Version 2 was laid out using an 8 bit ISA interface and ISA form factor. It added a PC/104 (<http://pc104.org/>) interface near the ISA connector. It also added 6 additional TTL level inputs using a DB-9S connector.

The board is constructed using mainly CMOS parts to reduce power consumption and to resolve end of life issues with some of the 74LS parts.

2.4 Version 3

Version 3 was laid out using just a PC/104 interface and a PC/104 form factor. The DCC test system signals use a board mounted 2x13 pin array that may be connected to a DB-25S cable using a standard ribbon cable. The additional 8 input signals used a board mounted 2x5 pin array.

The board used surface mount parts to save space. It also simplified the bus interface buffering to save space.

Sender Board Theory of Operation

3 Circuit Description

This section summarizes each of the major circuit blocks in the sender board. Please refer to the schematics shown below:

- Version 1 and 2: Sender V2 Schematic.pdf
 Version 1 does not support the 8 additional inputs provided by U604 and P602.
- Version 3: Sender V3 Schematic Send104E.pdf

Please refer to the appropriate schematic as you read the circuit description.

3.1 Block Diagram

Refer to page 1 of the schematic for a block diagram of the circuit.

3.2 Bus Interface

3.2.1 Version 1 and Version 2

Refer to page 2 of the schematic to see the bus interface portion of the circuit. This part of the circuit provides an 8 bit IBM PC style interface. The circuit itself is part of the dedicated portion of the PDS-601 prototype board. The signals come onto the board through edge card connector J201. The address signals (BA0-BA19) are buffered by U201, U202, and U203. The 8 data bus signals (BD0-BD7) are buffered by U206. The miscellaneous control signals are buffered by U204 and U205.

3.2.2 Version 3

The bus interface buffering was eliminated in version 3 because the bus PC/104 bus lengths are very short.

Board select PAL U302 was also replaced with 2 74AC138 decoders U1 and U20. 5 pin board select DIP switch SW301 was replaced with 8 solder bridges.

The interrupt output was removed for version 3 since it was never used by the software.

Sender Board Theory of Operation

3.3 Select & Parallel Port

Refer to page 3 of the schematic to see the select and parallel port portion of the circuit. U302 is part of the dedicated bus interface logic and is a PAL circuit that is programmed to provide chip selects and an overall board select. SW301 is used to select the IO port addresses for the sender board. The following addresses are supported:

<i>Address</i>	<i>SW1</i>	<i>SW2</i>	<i>SW3</i>	<i>SW4</i>	<i>Default</i>
240-257	Off	Off	On	Off	
260-277	Off	Off	On	On	
280-297	Off	On	Off	Off	
2A0-2B7	Off	On	Off	On	
2C0-2D7	Off	On	On	Off	
2E0-2F7	Off	On	On	On	
300-317	On	Off	Off	Off	
320-337	On	Off	Off	On	
340-357	On	Off	On	Off	X
360-377	On	Off	On	On	
380-397	On	On	Off	Off	
3A0-3B7	On	On	Off	On	
3C0-3D7	On	On	On	Off	
3E0-3F7	On	On	On	On	

U301 is an 8255 parallel port which is used to provide an 8 bit data port as well as miscellaneous input and output bits. This part is in the dedicated section of the board.

Port A of the 8255 is configured as a strobed output data port and holds the DCC output data. The control signals for port A are provided by port C described below.

Sender Board Theory of Operation

Port B is configured as an 8 bit input port and provides the following input signals:

<i>Bit</i>	<i>Name</i>	<i>Direction</i>	<i>Description</i>
PB0	UNDERFLOW	INPUT	Set to 1 if the board underflows.
PB1	-CLK1	INPUT	Readback of -CLK1 signal.
PB2	DCC0	INPUT	Readback of DCC0 signal.
PB3	DCCQH	INPUT	Readback of DCCQH signal.
PB4	82G0H	INPUT	Readback of 82G0H signal.
PB5	-SHOLD	INPUT	Readback of -SHOLD signal.
PB6	SCOPE	INPUT	Readback of SCOPE signal.
PB7	DCCOUTD	INPUT	Readback of DCCOUTD signal.

Port C is configured to provide the control signals for port A which is used as a strobed output port. The remaining signals are configured as outputs and support miscellaneous control functions. Pins **PC0 (PCRST)**, **PC1 (CPUEN)**, and **PC4 (-UNDERCLR)** are pulled up to make sure the board assumes a known state when it is reset.

<i>Bit</i>	<i>Name</i>	<i>Direction</i>	<i>Description</i>
PC0	PCRST	OUTPUT	Setting to 1 will reset the board.
PC1	CPUEN	OUTPUT	Setting to 1 activates software clock.
PC2	-CPUCLK	OUTPUT	Software -CLK1. A 1 to 0 transition will clock the board.
PC3	INTRA	OUTPUT	A 1 indicates port A needs a data byte.
PC4	-UNDERCLR	OUTPUT	Setting to 0 clears the UNDERFLOW bit
PC5	SCOPE	OUTPUT	Setting to 1 activates scope signal on the next DCC byte.
PC6	-ACKA	INPUT	A 0 indicates port A data byte is being loaded.
PC7	-OBFA	OUTPUT	A 0 indicates the port A data is available.

3.4 Clock Generators

Refer to page 4 of the schematic to see the clock generator portion of the circuit. The primary clock source for the board is **CLK1** and its inverse **-CLK1**. These signals are normally generated by 1 MHz. crystal oscillator U402. It is also possible for the master clock to be generated under software control by toggling **PC2 (-CPUCLK)**. The clock is generated by software if **PC1 (CPUEN)** is set to 1. Flip flop U403A and multiplex circuit U404A and U404B is used to switch between software and hardware clocks without a glitch. No glitch will occur as long as **-CPUCLK** is set to 0 prior to switching the state of **CPUEN**.

The heart of the clock generator circuit is the U401 82C54 programmable counter. Its **CLK0** output is used to generate a square wave signal on **82OUT0T** whenever the **DCCQH** data is 0. The duration of this signal is set to the duration of a DCC 0 pulse. The clock input to the counter is gated by U406A.

CLK1 is used to generate the high portion of a DCC 0 pulse. It begins counting at the same time **82OUT0T** begins a cycle and is programmed to produce a single pulse on **82OUT0H** at the point the **DCC0** signal should switch from 1 to 0. As such, **CLK0** sets the total duration of a DCC 0 pulse and **CLK1** sets the pulse high time. This allows stretched zeros to be generated.

Sender Board Theory of Operation

82OUT0T and **82OUT0H** are passed to flip flop U403B which generates the overall DCC 0 pulse signal **DCC0**.

Note: DCC0 must be initialized to 0 as part of the board initialization for proper operation. This is accomplished by forcing **82OUT0H** low temporarily which forces **DCC0** low.

CLK2 generates the DCC 1 signal **DCC1** in a manner similar to the way **CLK0** generates **82OUT0T**. Its clock is gated by U406B so that it only counts when DCC data signal **DCCQH** is 1. It always has a 50% duty cycle since pulse stretching is not needed for DCC 1 pulses.

AND gate U405B combines DCC0 and DCC1 to produce the complete DCC signal **DCCOUT**.

3.5 DCC Signal Generator

Refer to page 5 of the schematic to see the DCC signal generator portion of the circuit. U503A, U405C, and U406C are used to generate the **82G0H** signal used to synchronize the 82C54 **CLK1** signal which generates **82OUT0H**. **82G0H** will go low for one clock cycle at the beginning of a DCC 0 pulse and resets CLK1. It remains high during all other times.

DCC data is shifted out using shift register U501. U501 generates **DCCQH** which will be set to 0 to send a DCC 0 pulse, and 1 to send a DCC 1 pulse. It is shifted on each 0 to 1 transition of **DCCOUT** in order to make the next DCC data bit available on **DCCQH**. If no data is loaded into the shift register, it will eventually underflow and begin sending DCC 0 pulses. This is true because in serial input (SER) line of the shift register is tied to 0.

A new data BYTE is loaded into U501 under control of the state machine made up of U503B, U502, etc. In operation, a new data BYTE is sent to port A of the 8255. This causes PC7 (**-OBFA**) to go to 0 when the new data is present on the output port. **-OBFA** is synchronized to **-CLK1** by flip flop U503B. To prevent an underflow, data must be written to port A at least one half clock cycle (500 nsec.) plus setup time prior to the completion of the present BYTE.

Counter U502 together with gates U404C and U404D is used to load new data BYTES into the U501 shift register or signal an underflow at appropriate times. In normal operation, loading a new BYTE causes U502 to be set to an initial count of 8. It is incremented each time the **DCCOUT** signal goes from 0 to 1. As such, it maintains a count of each bit sent.

The main action occurs when 8 bits have been sent. At this point, U502 will be at count 15 and the carry out signal (**RCO**) will go to 1 when **DCCOUT** goes from 1 to 0 half way through the sending of the last bit. If a new data BYTE is available, **DATARDY** will be high at this time, causing the U404D output signal **-DATALD** to go to 0. Setting **-DATALD** to 0 causes a new data BYTE to be loaded into shift register U501 on the 0 to 1 transition of **DCCOUT**. It also resets counter U502 to count 8 and resets the **-OBFA** high via PC6 (**-ACKA**).

Note: -DATALD is also connected to the clear lead of U503B. This is done to prevent **DATARDY** from going inactive prematurely.

If a new data BYTE is not ready when it is needed, the U404C output signal **-SHOLD** will go low. This signal causes the U502 counter to remain at count 15. Also, U501 will shift a 0 out on **DCCQH**, causing a DCC 0 pulse to be sent. Flip flop U504A produces the latched **PB0 (UNDERFLOW)** signal. Once activated, the **UNDERFLOW** signal remains at 1 until it is reset from software using the PC4 (**-UNDERCLR**) signal. Software should test the **UNDERFLOW** signal periodically to make sure no data BYTES were missed.

The **SCOPE** signal is generated by U406D. The **SCOPE** output will follow the state of **-DATALD** whenever software sets PC5 (**-SCOPE**) low prior to sending a BYTE of data. This causes a single 1 pulse to occur on the **SCOPE** line with a duration equal to one half the pulse width of the last DCC pulse sent. This pulse will occur just prior to the beginning of a new BYTE of DCC data pulses.

Note: The external DCC signal is actually **DCCOUTD** which is delayed one **CLK1** time (1 microsecond) from **DCCOUT**. This gives enough time for a scope to trigger from **SCOPE** and still show a view of the first 0 to 1 transition of the selected DCC output BYTE.

Sender Board Theory of Operation

3.6 External Interface

3.6.1 RS-422 Outputs and Inputs

Refer to page 6 of the schematic to see the external interface portion of the circuit. The two external output signals **DCCOUTD** and **SCOPE** are buffered by RS-422 driver U601. These balanced signals will drive most types of boosters or scopes and provide excellent noise immunity.

U602 is an RS-422 balanced receiver which provides 4 balanced inputs. These input bits can be read by the software by issuing an IN command. Each complementary input lead is pulled down by a 10K resistor to make sure that unused leads remain at 1.

Note: It is assumed that the balanced inputs to the sender board will be used, possibly with some external hardware, to convey motor direction, lamp status, or accessory state to the sender board. The software will use this data as part of the receiver testing process.

3.6.2 6 TTL Level Outputs

6 TTL level outputs are provided in addition to the RS-422 signals. Bus register U603 clocks the bus signals to AUXOUT0 through AUXOUT5 on the rising edge of -AUXOUT . On power up, the outputs are cleared by the -PCRST signal.

3.6.3 8 TTL Signal Inputs (Version 2 & 3 Only)

The 8 TTL input signals are connected to the inputs of Tri-State buffer U604. Each input is pulled up to +5 Volts by 10K SIP resistor array RS601. The signals are fed to the data bus when the -AUXIN is driven low.