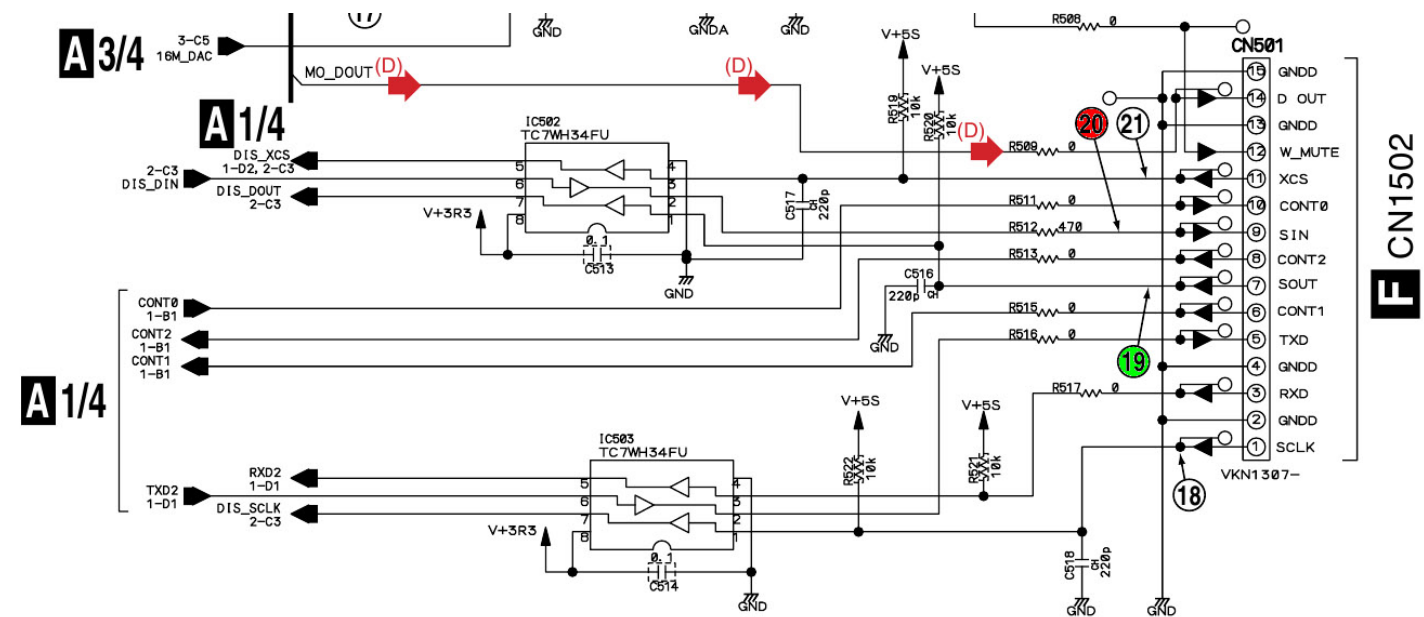


Pioneer CDJ-1000mk3 Service Manual:



19 - Display Board => Main Assy
20 - : Main Assy => Display Board

Data: Main Assy => Display Board @ Pioneer CDJ-1000mk3

Example: MSB b7.....b0 LSB

/	0byte	1byte	2byte	3byte	4byte	5byte	6byte	7byte	8byte	9byte	10byte	11byte	12byte	13byte	14byte	15byte	16byte	17byte	18byte	19byte	20byte	21byte	22byte	23byte	24byte	25byte	26byte
0 package	0	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	POS	CRC
1 package	24	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	POS	CRC
2 package	48	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	POS	CRC
3 package	72	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	WFM	POS	CRC
4 package	96	WFM	WFM	WFM	WFM	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	00000000	BAR	BAR	BAR	BAR	BAR	BAR	POS	CRC
5 package	120	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	BAR	POS	CRC
6 package	144	ALL	ALL	ALL	AA000000	ALL	ALL	ALL	ALL	ALL	AA000000	ALL	ALL	ALL	ALL	ALL	ALL	ALL	A10001A1	00000000	00000000	00000000	00000000	00000000	00000000	POS	CRC
7 package	168	ALL	ALL	ALL	ALL	ALL	ALL	ALL	ALL	AAA00001	1AA10000	ALL	AA000000	00000000	00000000	00000000	00000000	ALL	0AAAAAA	AAAAA000	ALL	ALL	00000000	0AAAA000	00000000	POS	CRC

[all packages][0 byte] – number of package
[all packages][25 byte] = POS 1...135 – jog display position cursor 138 = eject animation 137 = load in animation 136 = fill circle on display
[all packages][26 byte] – CRC = (1byte + 2byte + ... + 25byte)%256
BAR - display TIME progress bar (start: [4package][19 byte] end: [5package][8 byte] first bit: b0)
BAR - display CUE progress bar (start: [5package][9 byte] end: [5package][21 byte] first bit: b0)
BAR - display MEMORY progress bar (start: [4package][5 byte] end: [4package][17 byte] first bit: b0)
WFM – display WAVEFORM DATA (start: [0package][1 byte] end: [4package][4 byte] first bit: b0)

[6package][1 byte] = xA Track Number (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[6package][2 byte] = Ax Track Number or xA Folder Number
[6package][3 byte] = Ax Folder Number (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)

[6package][4 byte][bit6] = ‘FLD’ on display
[6package][4 byte][bit7] = ‘TRACK’ on display

[6package][5 byte] [bit0] = 11 number of track on display table
[6package][5 byte] [bit1] = 21 number of track on display table
[6package][5 byte] [bit2] = 31 number of track on display table
[6package][5 byte] [bit3] = 41 number of track on display table
[6package][5 byte] [bit4] = 11 track enable CUE in memory on display table

[6package][5 byte] [bit5] = **[2]** track enable CUE in memory on display table
[6package][5 byte] [bit6] = **[3]** track enable CUE in memory on display table
[6package][5 byte] [bit7] = **[4]** track enable CUE in memory on display table

[6package][6 byte] [bit0] = **[5]** number of track on display table
[6package][6 byte] [bit1] = **[6]** number of track on display table
[6package][6 byte] [bit2] = **[7]** number of track on display table
[6package][6 byte] [bit3] = **[8]** number of track on display table
[6package][6 byte] [bit4] = **[5]** track enable CUE in memory on display table
[6package][6 byte] [bit5] = **[6]** track enable CUE in memory on display table
[6package][6 byte] [bit6] = **[7]** track enable CUE in memory on display table
[6package][6 byte] [bit7] = **[8]** track enable CUE in memory on display table

[6package][7 byte] [bit0] = **[13]** track enable CUE in memory on display table
[6package][7 byte] [bit1] = **[14]** track enable CUE in memory on display table
[6package][7 byte] [bit2] = **[15]** track enable CUE in memory on display table
[6package][7 byte] [bit3] = **[16]** track enable CUE in memory on display table
[6package][7 byte] [bit4] = **[13]** number of track on display table
[6package][7 byte] [bit5] = **[14]** number of track on display table
[6package][7 byte] [bit6] = **[15]** number of track on display table
[6package][7 byte] [bit7] = **[16]** number of track on display table

[6package][8 byte] [bit0] = **[9]** number of track on display table
[6package][8 byte] [bit1] = **[10]** number of track on display table
[6package][8 byte] [bit2] = **[11]** number of track on display table
[6package][8 byte] [bit3] = **[12]** number of track on display table
[6package][8 byte] [bit4] = **[9]** track enable CUE in memory on display table
[6package][8 byte] [bit5] = **[10]** track enable CUE in memory on display table
[6package][8 byte] [bit6] = **[11]** track enable CUE in memory on display table
[6package][8 byte] [bit7] = **[12]** track enable CUE in memory on display table

[6package][9 byte] [bit0] = **[17]** track enable CUE in memory on display table
[6package][9 byte] [bit1] = **[18]** track enable CUE in memory on display table
[6package][9 byte] [bit2] = **[19]** track enable CUE in memory on display table
[6package][9 byte] [bit3] = **[20]** track enable CUE in memory on display table
[6package][9 byte] [bit4] = **[17]** number of track on display table
[6package][9 byte] [bit5] = **[18]** number of track on display table
[6package][9 byte] [bit6] = **[19]** number of track on display table
[6package][9 byte] [bit7] = **[20]** number of track on display table

[6package][10 byte] [bit6] =CUE> on display
[6package][10 byte] [bit7] =MEMORY> on display

[6package][11 byte] = xA Time: SEC (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[6package][12 byte] = Ax Time: SEC (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[6package][13 byte] = xxA Time: MIN (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[6package][14 byte] = xAx Time: MIN (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[6package][15 byte] = Axx Time: MIN (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[6package][16 byte] = xxA ALL TRACKS (119 = ‘0’ 93 = ‘5’ 18 = ‘1’) when tracks>99 only
[6package][17 byte] = xA ALL TRACKS (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[6package][18 byte] = Ax ALL TRACKS (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)

[6package][19 byte][bit1] = REMAIN on display
[6package][19 byte][bit7] = A.CUE on display

[7package][1 byte] = xxA tempo (119 = ‘0’ 93 = ‘5’)
[7package][2 byte] = xAx tempo (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[7package][3 byte] = Axx tempo (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[7package][4 byte] = xA Time: Frame (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[7package][5 byte] = Ax Time: Frame (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[7package][6 byte] = xxA BPM (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)
[7package][7 byte] = xAx BPM (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)

[7package][8 byte] = Axx BPM (119 = ‘0’ 93 = ‘5’ 18 = ‘1’)

[7package][9 byte][]
[7package][9 byte]: x11xxxxx (‘+’ on tempo)
 x10xxxxx (‘-’ on tempo)
 x00xxxxx (‘ ’ on tempo)

[7package][9 byte][bit7] = Ax.xx% = ‘1’ of pitch display

[7package][10 byte] [bit5] = xx.x% = ‘.’ dot of pitch display
[7package][10 byte] [bit6] = x.xx% = ‘.’ dot of pitch display

[7package][11 byte] : 10010000 +-6% on display
 10100000 +-10% on display
 11000000 +-16% on display
 10000000 **WIDE** on display

[7package][12 byte][bit6] = 1 when WIDE Pitch Enable
[7package][12 byte][bit7] – MT on display or red diode on button

[7package][17 byte][bit0] = **PLAY LED**
[7package][17 byte][bit1] = **CUE LED**
[7package][17 byte][bit2] = **LOOP IN LED**
[7package][17 byte][bit3] = **LOOP OUT LED**
[7package][17 byte][bit4] = **RELOOP EXIT LED**
[7package][17 byte][bit5] = **RED DIODE REVERSE**
[7package][17 byte][bit6] = **RED DIODE HOT CUE A**
[7package][17 byte][bit7] = **GREEN DIODE HOT CUE A**

[7package][18 byte][bit0] = **ORANGE DIODE HOT CUE A**
[7package][18 byte][bit1] = **RED DIODE HOT CUE B**
[7package][18 byte][bit2] = **GREEN DIODE HOT CUE B**
[7package][18 byte][bit3] = **ORANGE DIODE HOT CUE B**
[7package][18 byte][bit4] = **RED DIODE HOT CUE C**
[7package][18 byte][bit5] = **GREEN DIODE HOT CUE C**
[7package][18 byte][bit6] = **ORANGE DIODE HOT CUE C**
[7package][18 byte][bit7] = 0

[7package][19 byte][bit3] =MT red diode on button
[7package][19 byte][bit4] = tempo reset green diode
[7package][19 byte][bit5] = **Vinyl blue diode**
[7package][19 byte][bit6] = **CDJ green diode**
[7package][19 byte][bit7] = **SD card green diode**

[7package][20 byte] : 00001000 +-6% on display
 00011000 +-10% on display
 00111000 +-16% on display
 01111000 **WIDE** on display

[7package][21 byte] CPOS 1...85 – jog display position CUE cursor 0 = without cue on jog display

[7package][23 byte][bit3] [bit4] = ”Vinyl ON” on VFD lamp
[7package][23 byte][bit5] = circle “touch enable” on VFD lamp
[7package][23 byte][bit6] = circle “memory empty” on VFD lamp

7-segment code:

119 = ‘0’
18 = ‘1’
107 = ‘2’
91 = ‘3’
30 = ‘4’

93 = ‘5’
125 = ‘6’
19 = ‘7’
127 = ‘8’
95= ‘9’

Data: **Display Board => Main Assy @ Pioneer CDJ-1000mk3**

/	0byte	1byte	2byte	3byte	4byte	5byte	6byte	7byte	8byte	9byte	10byte	11byte	12byte	13byte	14byte	15byte	16byte	17byte	18byte	19byte	20byte	21byte	22byte	23byte	24byte	25byte	26byte
0-7 packages	1	16	ADC T/B	ADC R/S	ADC P	ADC P	ADCT	ADCT	PLS CNT	PLS CNT	JOG SPD	JOG SPD	JOG STS	00000000	ALL	00000000	ALL	ALL	ALL	CRC	241	100	118	00000000	00000000	00000000	00000000

[0-7packages][2 byte] = ADC data TOUCH/BRAKE 0...255 8 bits resolution
[0-7packages][3 byte] = ADC data RELEASE/START 0...255 8 bits resolution
[0-7packages][4 byte] = ADC Pitch MSB [AAAAAAA] 0...16383 14 bits resolution
[0-7packages][5 byte] = ADC Pitch LSB [AAAAAAA0]
[0-7packages][6 byte] = ADC Pitch Center Potentiometer MSB [AAAAAAA] 0...16383 14 bits resolution
[0-7packages][7 byte] = ADC Pitch Center Potentiometer LSB [AAAAAAA0]
[0-7packages][8 byte] = Jog Pulse Counter MSB 0...65535 16 bits resolution (3600 pulses per 1 round)
[0-7packages][9 byte] = Jog Pulse Counter LSB
[0-7packages][10 byte] = Jog Speed MSB 0...65535 16 bits resolution (when jog stopped speed = 65535)
[0-7packages][11 byte] = Jog Speed LSB

[0-7packages][12 byte][bit0] = Eject Lock switch: 1 = unlock / 0 = lock
[0-7packages][12 byte][bit1] = Direction switch: 1 = forward / 0 = reverse
[0-7packages][12 byte][bit2] = 0
[0-7packages][12 byte][bit3] = 0
[0-7packages][12 byte][bit4] = jog touch enable
[0-7packages][12 byte][bit5] = jog touch enable
[0-7packages][12 byte][bit6] – 1 = jog forward rotation / 0 = jog reverse rotation
[0-7packages][12 byte][bit7] = jog rotation detect

[0-7packages][14 byte][bit0] = PLAY button
[0-7packages][14 byte][bit1] = CUE button
[0-7packages][14 byte][bit2] = Loop IN button
[0-7packages][14 byte][bit3] = Loop OUT button
[0-7packages][14 byte][bit4] = Re loop/Exit button
[0-7packages][14 byte][bit5] = Hot Cue A button
[0-7packages][14 byte][bit6] = Hot Cue B button
[0-7packages][14 byte][bit7] = Hot Cue C button

[0-7packages][16 byte][bit0] = REC MODE
[0-7packages][16 byte][bit1] = << Track search
[0-7packages][16 byte][bit2] = Track search >>
[0-7packages][16 byte][bit3] = << Search
[0-7packages][16 byte][bit4] = Search >>
[0-7packages][16 byte][bit5] = 0
[0-7packages][16 byte][bit6] = 0
[0-7packages][16 byte][bit7] = 0

[0-7packages][17 byte][bit0] = < MP3 FOLDER SEARCH
[0-7packages][17 byte][bit1] = MP3 FOLDER SEARCH >
[0-7packages][17 byte][bit2] = JOG MODE button
[0-7packages][17 byte][bit3] = TEMPO button
[0-7packages][17 byte][bit4] = MASTER TEMPO button
[0-7packages][17 byte][bit5] = TEMPO RESET button
[0-7packages][17 byte][bit6] = 0
[0-7packages][17 byte][bit7] = 0

[0-7packages][18 byte][bit0] = CALL >
[0-7packages][18 byte][bit1] = < CALL

[0-7packages][18 byte][bit2] = MEMORY button
[0-7packages][18 byte][bit3] = DELETE
[0-7packages][18 byte][bit4] = EJECT button
[0-7packages][18 byte][bit5] = TIME MODE/AUTO CUE button
[0-7packages][18 byte][bit6] = TEXT MODE button
[0-7packages][18 byte][bit7] = 0

[0-7packages][19 byte] - CRC

Author: Anatsko Andrei