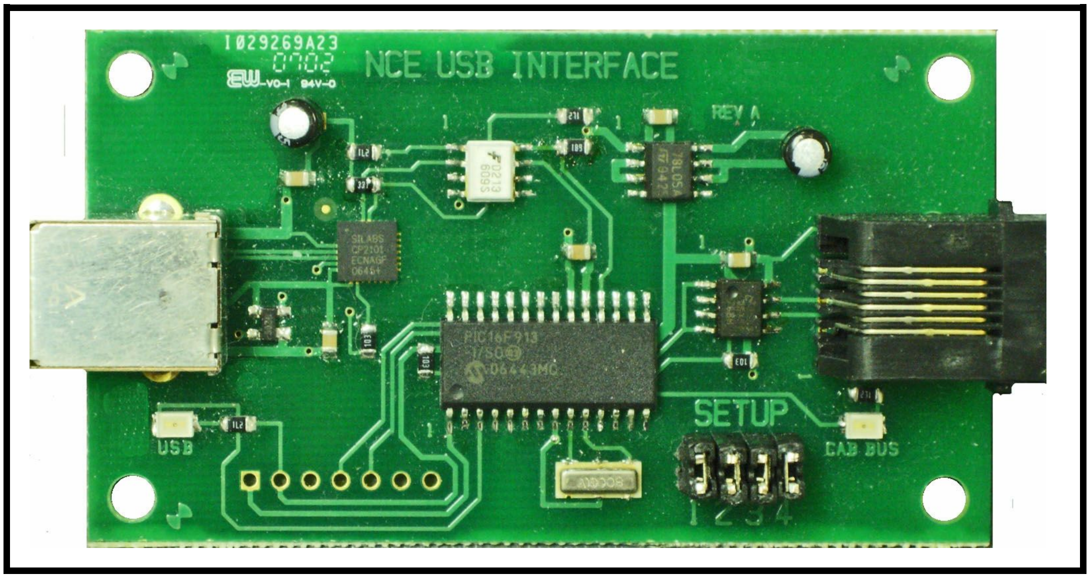
NCE The Power of DCC

USB

Interface board



$49.95

USB interface for NCE Cab Bus.

Features of the USB Interface Board:

- ✓ Supports train/accessory/signal operation and Ops programming with all NCE DCC systems
- ✓ Also Supports ***Power Cab*** program track
- ✓ Optoisolated USB prevents “sneak path” short circuits
- ✓ Multiple USB boards can be used with some NCE systems
- ✓ USB drivers for Windows, Mac OS X and Linux via Internet download

Use of this product ***requires*** Internet access to download the latest USB drivers

05240223

Webster, NY 14580

Description:

The USB interface board provides a means of connecting a computer to the NCE cab bus. While primarily designed to provide a computer interface to the Power Cab program track, it can additionally be used with any NCE system to run trains, control signals, turnouts and Program on the Main (OPs mode programming).

When used with the Power Cab the USB board cab bus address must be 3. When the appropriate shunts are installed to configure the USB board for use with a Power Cab the address is automatically forced to cab bus address 3. When configured for a different NCE system the cab bus address can be changed via a command (binary command 0xB1) through the USB port.

Optical isolation is provided to prevent “sneak path” short circuits when connected to a computer.

You may use as many USB interface boards as you wish, up to the limit of available cab addresses on your DCC system. Each USB board draws only 45mA from the Cab Bus power.

USB Driver Installation:

We recommend having the USB driver for your operating system available ***before*** connecting the USB interface board to your computer system. Due to the constant change of drivers for computer products we do not provide a disc with drivers for your computer. We do keep our website up to date with the latest drivers and driver installation manual. You can download these from the NCE website (www.ncedcc.com/usb_drivers).

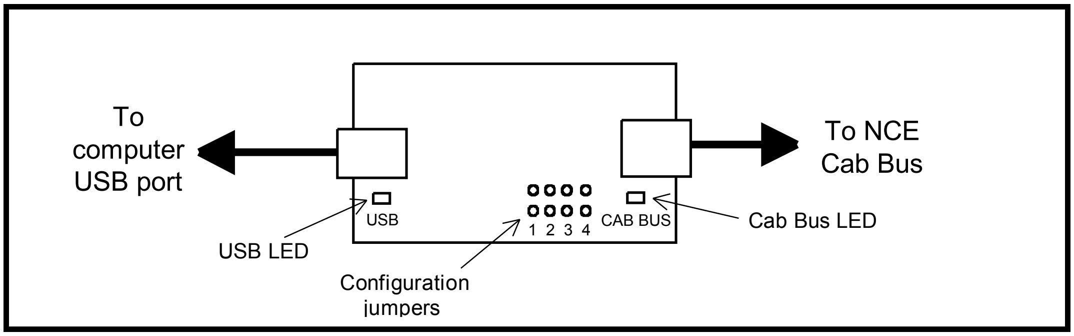
Note: If installing on a Win XP or Vista machine you can also let the Windows driver installation program search the Internet for the proper driver.

Mechanical Installation:

The USB interface has 4 mounting holes that can be used to mount it to a convenient place on your layout. We recommend using #4 x 3/4” round head wood screws with #4 flat washers. *Do not over tighten the screws.* Lightly snug the screws up just enough to keep the circuit board in place without bending it. Bending the circuit board can fracture the fragile ceramic electronic components.

Electrical Connections:

The diagram below illustrates the wiring of the USB interface between your computer and DCC system. The computer connection is via Type A to Type B USB cable. Connection to the NCE Cab Bus is via the supplied modular cable. This product will only work with NCE DCC system products.



Configuration jumpers for USB board

JP1 off = 9600 baud
JP1 on = 19200 baud

Jumpers matching the system/version table below MUST be installed for proper operation. Some commands are “massaged” by the USB board to accommodate bugs in different devices so the board needs to know what system it is connected to by reading the jumpers.

Install the included shunts (jumpers) to match your DCC system.

System and Version (as read on cab LCD)	JP2	JP3	JP4	Version reported to computer by 0xaa command
Power Cab v1.28c	-	-	-	6.3.0
SB3 v1.28b	-	-	ON	6.3.1
PH Pro 2007	-	ON	-	6.3.2
All Sys[1]	-	ON	ON	6.3.3
Future Versions				
Power Cab	ON	-	-	6.3.4
SB3	ON	-	ON	6.3.5
unused	ON	ON	-	6.3.6
unused	ON	ON	ON	6.3.7

Notes:

1) “All Sys” means the USB board makes no distinction between what version system you are using and passes all commands from the computer to the USB board
2) The third byte returned by the version command (0xaa) is the “bit-reversed” status of jumpers 2,3 and 4 (see table above).

Technical things you should know:

- The USB board forces the cab bus address to 3 when used with a PowerCab.
- The USB board cab bus address and baud rate is initialized only at power up. All board power comes from the cab bus connection so the board must be power cycled via unplugging from the cab bus if the address is changed.
- Cab bus addresses for the SB3 are limited to a range of 2 through 5.
- Multiple USB boards may be used with a system (except Power Cab) for control by multiple computers. Each USB board must have a different cab address ranging from 2 to 63 on Power Pro systems and 2-5 on SB3 systems.

Beyond this point is technical stuff you probably don’t need to read unless you are writing your own software

Commands supported by PowerCab-1.28 (no jumpers)

Hex	Bytes	Description
0x80	1 bytes	NOP... just returns '!'
0x8C	1 bytes	Returns !,CR,LF
0x9C	2 bytes	Execute route macro
0x9E	1 bytes	Enter program track mode
0x9F	1 bytes	Exit program track mode
0xA0	4 bytes	Write a CV in paged mode
0xA1	3 bytes	Read a CV in paged mode
0xA2	5 bytes	Locomotive control command
0xA6	3 bytes	Write in register mode
0xA7	2 bytes	Read in register mode
0xA8	4 bytes	Write in direct mode
0xA9	3 bytes	Read in direct mode
0xAA	1 bytes	Return C/S software version (USB board = 6.3.x - see table for value of x)
0xAD	5 bytes	Accy/signal/macro commands (NOTE: accy/sig addresses limited to range of 1-250)
0xAE	6 bytes	Locomotive OPs program byte
0xAF	6 bytes	Accy/Signal OPs program byte (NOTE: accy/sig addresses limited to range of 1-250)
0xB0	5 bytes	reserved - factory test

Notes: jumpers MUST be installed to match

Commands supported by SB3-1.28 - JP4 only

Hex	Bytes	Description
0x80	1 bytes	NOP... just returns '!'
0x8C	1 bytes	Return !,CR,LF
0x9C	2 bytes	Execute route macro
0xA2	5 bytes	Locomotive control command
0xAA	1 bytes	Return C/S software version
0xAD	5 bytes	Accy/signal/macro commands (NOTE: accy/sig addresses limited to range of 1-250)
0xAE	6 bytes	Locomotive OPs program byte
0xAF	6 bytes	Accy/Signal OPs program byte (NOTE: accy/sig addresses limited to range of 1-250)
0xB0	5 bytes	reserved - factory test
0xB1	2 bytes	Set cab address of USB device

Commands supported by Power Pro 2007 version - JP3 only

Hex	Bytes	Description
0x80	1 bytes	NOP... just returns '!'
0x8C	1 bytes	Returns !,CR,LF
0x9C	2 bytes	Execute route macro
0xA2	5 bytes	Locomotive control command
0xAA	1 bytes	Return C/S software version
0xAD	5 bytes	Accy/signal/macro commands (no limit on accy/sig addresses)
0xAE	6 bytes	not supported due to bug in command station software
0xAF	6 bytes	not supported due to bug in command station software
0xB0	5 bytes	reserved - factory test
0xB1	2 bytes	Set cab bus address of this USB device (this command only works internally to the USB interface it does not get to the command station)

Commands supported by ALLSYS - JP3 and JP4

Hex	Bytes	Description
0x80	1 bytes	NOP... just returns '!'
0x8C	1 bytes	Returns !,CR,LF
0x9C	2 bytes	Execute route macro
0x9E	1 bytes	Enter program track mode
0x9F	1 bytes	Exit program track mode
0xA0	4 bytes	Write a CV in paged mode
0xA1	3 bytes	Read a CV in paged mode
0xA2	5 bytes	Locomotive control command
0xA6	3 bytes	Write in register mode
0xA7	2 bytes	Read in register mode
0xA8	4 bytes	Write in direct mode
0xA9	3 bytes	Read in direct mode
0xAA	1 bytes	Return C/S software version
0xAD	5 bytes	Accy/signal/macro commands
0xAE	6 bytes	Locomotive OPs program byte
0xAF	6 bytes	Accy/Signal OPs program byte
0xB0	5 bytes	reserved - factory test
0xB1	2 bytes	Set cab bus address of this USB device (this command only works internally to the USB interface it does not get to the command station)

A list of the expected number of bytes (by the USB) for all binary commands

Command	Bytes		Command	Bytes		Command	Bytes
0x80	1		0x93	5		0xA6	3
0x81	4		0x94	6		0xA7	2
0x82	1		0x95	7		0xA8	4
0x83	1		0x96	8		0xA9	3
0x84	1		0x97	4		0xAA	1
0x85	3		0x98	5		0xAB	1
0x86	2		0x99	7		0xAC	1
0x87	2		0x9A	11 dec		0xAD	5
0x88	3		0x9B	2		0xAE	6
0x89	1		0x9C	2		0xAF	6
0x8A	2		0x9D	3		0xB0	5
0x8B	1		0x9E	1		0xB1	2
0x8C	1		0x9F	1		0xB2	1
0x8D	4		0xA0	4		0xB3	1
0x8E	20 dec		0xA1	3		0xB4	1
0x8F	3		0xA2	5		0xB5	1
0x90	18 dec		0xA3	4		0xB6	1
0x91	18 dec		0xA4	5		0xB7	1
0x92	10 dec		0xA5	6		0xB8	1

Unsupported commands

The following commands are completely unsupported by the current version of the USB board. The USB board will buffer the expected number of bytes (1 byte for unsupported commands) then return ASCII '0' indicating the command is not supported.

0x81 -> 0x8B
0x8D -> 0x9B
0x9D
0xA3
0xA4
0xA5
0xAB
0xAC
0XB3 -> 0XBF

Description of command formats used with NCE RS232 and USB interfaces

```
;*************************************************************************
;
;              The RS-232 port binary commands are designed to work
;              in a computer friendly mode.
;
;              Command format is:  <cmd number> <data> <data> ...
;
;              Commands range from 0x80 to 0xBF
;
;
;          NOTE: For commands 0x9F,0xA0,0xA1 and 0xA6- 0xA9 a single
;                byte of 0 will be returned if not in programming mode
;
;
;
;              Errors returned:    '0'= command not supported
;                                  '1'= loco/accy/signal address out of range
;                                  '2'= cab address or op code out of range
;                                  '3'= CV address or data out of range
;                                  '4'= byte count out of range
;                                  '!'= command completed successfully
;
;*************************************************************************
;
; CMD  FORMAT   DESCRIPTION                # OF BYTES   RESPONSES
;                                            RETURNED
;--------------------------------------------------------------------------
; 0x80          NOP, dummy instruction           (1)     !
;--------------------------------------------------------------------------
; 0x81 xx xx yy assign loco                      (1)     !,1,2
;--------------------------------------------------------------------------
; 0x82          read clock                       (2)     <hours>
;                                                        <minutes>
;--------------------------------------------------------------------------
; 0x83          Clock stop                       (1)     !
;--------------------------------------------------------------------------
; 0x84          Clock start                      (1)     !
;--------------------------------------------------------------------------
; 0x85 xx xx    Set clock hr/min                 (1)     !,3
;--------------------------------------------------------------------------
; 0x86 xx       Set clock 12/24                  (1)     !,3
;               0=12 hr 1=24 hr
;--------------------------------------------------------------------------
; 0x87 xx       Set clock ratio                  (1)     !,3
;--------------------------------------------------------------------------
; 0x88 xx xx    Dequeue packet by loco addr      (1)     !,1,2
;--------------------------------------------------------------------------
; 0x89          Enable main trk, kill prog       (1)     !
;--------------------------------------------------------------------------
; 0x8A yy       Return status of AIU yy          (4)     <current hi byte>
;                                                        <current lo byte>
;                                                        <change hi byte>
;                                                        <change lo byte>
;--------------------------------------------------------------------------
; 0x8B          Kill main trk, enable prog       (1)     !
;--------------------------------------------------------------------------
; 0x8C          Dummy instruction returns "!"    (3)     !,0x0D,0x0A
;               followed by CR/LF
;--------------------------------------------------------------------------
; 0x8D xxxx mm  Set speed mode of loco xxxx      (1)     !,1,3
;               to mode mm, 1=14, 2=28, 3=128            <speed mode, 0 to 3>
;--------------------------------------------------------------------------
; 0x8E aaaa nn  <16 data bytes>                   (1)     !,4
;               Write nn bytes, start at aaaa
;               Must have 16 data bytes, pad
;               them out to 16 if necessary
;--------------------------------------------------------------------------
; 0x8F aaaa     Read 16 bytes, start at aaaa     (16)    16 bytes
;--------------------------------------------------------------------------
; 0x90 cc xx... Send 16 char message to Cab cc   (1)     !,2
;               LCD line 3.  xx = 16 ASCII char
;--------------------------------------------------------------------------
```

```
;-------------------------------------------------------------------------
; 0x91 cc xx... Send 16 char message to Cab cc   (1)     !,2
;               LCD line 4.  xx = 16 ASCII char
;-------------------------------------------------------------------------
; 0x92 cc xx... Send 8 char message to Cab cc    (1)     !,2
;               LCD line 2 right.  xx = 8 char
;-------------------------------------------------------------------------
; 0x93 ss xx xx xx                               (1)     !
;               Queue 3 byte packet to TEMP_Q
;               Send ss times
;-------------------------------------------------------------------------
; 0x94 ss xx xx xx xx                            (1)     !
;               Queue 4 byte packet to TEMP_Q
;               Send ss times
;-------------------------------------------------------------------------
; 0x95 ss xx xx xx xx xx                         (1)     !
;               Queue 5 byte packet to TEMP_Q
;               Send ss times
;-------------------------------------------------------------------------
; 0x96 ss xx xx xx xx xx xx                      (1)     !
;               Queue 6 byte packet to TEMP_Q
;               Send ss times
;-------------------------------------------------------------------------
; 0x97 aaaa xx  Write 1 byte,to aaaa             (1)     !
;-------------------------------------------------------------------------
; 0x98 aaaa xx xx                                (1)     !
;               Write 2 bytes to aaaa
;-------------------------------------------------------------------------
; 0x99 aaaa <4 data bytes>                       (1)     !
;               Write 4 bytes to aaaa
;-------------------------------------------------------------------------
; 0x9A aaaa <8 data bytes>                       (1)     !
;               Write 8 bytes to aaaa
;-------------------------------------------------------------------------
; 0x9B yy       Return status of AIU yy          (2)     <current hi byte>
;               (short form of command 0x8A)             <current lo byte>
;-------------------------------------------------------------------------
; 0x9C xx       Execute macro number xx          (1)     !,0,3
;-------------------------------------------------------------------------
; 0x9D aaaa     Read 1 bytes from aaaa           (1)     1 byte
;-------------------------------------------------------------------------
; 0x9E          Enter Programming track mode     (1)     ! = success
;                                                        3 = short circuit
;-------------------------------------------------------------------------
; 0x9F          Exit Programming track mode      (1)     ! = success
;-------------------------------------------------------------------------
; 0xA0 aaaa xx  Program CV aa with data xx in    (1)     ! = success
;               paged mode                               0 = program track no enabled
;-------------------------------------------------------------------------
; 0xA1 aaaa     Read CV aa in paged mode         (2)     !,0,3
;               NOTE: cv data followed ! for ok,
;                     0xff followed by 3 for can't read
;-------------------------------------------------------------------------
; 0xA2 <4 data bytes>
;                Locomotive control command      (1)      !,1
;
;                Sends a speed or function packet to a locomotive.
;                Command Format:  0xA2 <addr_h> <addr_l> <op_1> <data_1>
;
;                Addr_h and Addr_l are the loco address in DCC format.
;                If a long address is in use, bits 6 and 7 of the high byte are set.
;                  Ex:  Long address  3 = 0xc0 0x03
;                       Short address 3 = 0x00 0x03
;
;
;                 op_1       data_1    Operation description
;
;                 01         0-7f      Reverse 28 speed command
;                 02         0-7f      Forward 28 speed command
;                 03         0-7f      Reverse 128 speed command
;                 04         0-7f      Forward 128 speed command
;                 05         0         Estop reverse command
;                 06         0         Estop forward command
;                 07         0-1f      Function group 1, bits 0-3 = F1-F4, bit 4=F0
;                 08         0-0f      Function group 2, bits 0-3 = F5-F8
;                 09         0-0f      Function group 3, bits 0-3 = F9-F12
;                 0a         0-7f      Set reverse consist address for lead loco
;                 0b         0-7f      Set forward consist address for lead loco
;                 0c         0-7f      Set reverse consist address for rear loco
;                 0d         0-7f      Set forward consist address for rear loco
;                 0e         0-7f      Set reverse consist address for additional loco
;                 0f         0-7f      Set forward consist address for additional loco
;                 10         0         Del loco from consist
;                 11         0         Kill consist
;                 12         0-9       Set momentum
;                 13         0-7f      No action, always returns success
;                 14         0-7f      No action, always returns success
;                 15         0-ff      Functions 13-20 control (bit 0=F13, bit 7=F20)
;                 16         0-ff      Functions 21-28 control (bit 0=F21, bit 7=F28)
;                 17-7f      reserved  reserved
;
;--------------------------------------------------------------------------
; 0xA3 xx xx    Queue 3 byte packet to TRK_Q     (1)     !,1
;               (replaces any packet with same
;               address if it exists)
;--------------------------------------------------------------------------
; 0xA4 xx xx... Queue 4 byte packet to TRK_Q     (1)     !,1
;               (replaces any packet with same
;               address if it exists)
;--------------------------------------------------------------------------
; 0xA5 xx xx... Queue 5 byte packet to TRK_Q     (1)     !,1
;               (replaces any packet with same
;               address if it exists)
;--------------------------------------------------------------------------
; 0xA6 rr xx    Program register rr with data xx (1)     ! = success
;               in register mode                         0 = program track no enabled
;--------------------------------------------------------------------------
; 0xA7 rr       Read register rr in register mode(2)     !,3
;               NOTE: cv data followed ! for ok,         0 = program track no enabled
;                     0xff followed by 3 for can't read
;--------------------------------------------------------------------------
; 0xA8 aaaa xx  Program CV aaaa with data xx     (1)     ! = success
;               in direct mode                           0 = program track no enabled
;--------------------------------------------------------------------------
; 0xA9 aaaa     Read CV aaaa in direct mode      (2)     !,3
;               NOTE: cv data followed ! for ok,         0 = program track no enabled
;                     0xff followed by 3 for can't read
;--------------------------------------------------------------------------
; 0xAA          Return software revision number  (3)     <data1>,<data2>,<data3>
;               FORMAT: VV.MM.mm
;--------------------------------------------------------------------------
; 0xAB          Perform a soft reset of command  (0)     Returns nothing
;               station (like cycling the power)
;--------------------------------------------------------------------------
; 0xAC          Perform a hard reset of command  (0)     Returns nothing
;               station (reset to factory defaults)
;--------------------------------------------------------------------------
; 0xAD <4 data bytes>
;               Accy/signal and macro commands     (1)     !,1
;
;               Command Format:  0xAD <addr_h> <addr_l> <op_1> <data_1>
;
;               Addr_h and Addr_l are the accessory/signal address (NOT in DCC format).
;                 Ex:  Accessory Address  513 = 0x02 0x01  (hi byte first)
;
;               NOTE: accy/signal address 0 is not a valid address
;
;                Op_1    Data_1       Operation description
;
;                01         0-255     NCE macro number 0-255
;                02         0-255     Duplicate of Op_1 command
;                03         0        Accessory  Normal direction (ON)
;                04         0        Accessory  Reverse direction (OFF)
;                05         0-1f      Signal Aspect 0-31
;                05-7f      reserved  reserved
;
;--------------------------------------------------------------------------
```

```
;--------------------------------------------------------------------------
; 0xAE  <5 data bytes>
; ***USB COMMAND ONLY***
;               OPs program loco CV                (1)     !,0,1,3
;
;               Command Format: 0xA2 <addr_h> <addr_l> <CV_h> <CV_l> <data>
;
;               addr_h,addr_l are loco address (same as 0xA2 command)
;               CV_h, CV_l are cv address high byte first
;              data is 8 bit data for CV
;--------------------------------------------------------------------------
; 0xAF  <5 data bytes>
; ***USB COMMAND ONLY***
;               OPs program accessory/signal       (1)     !,0,1,3
;
;               Command Format: 0xA2 <addr_h> <addr_l> <CV_h> <CV_l> <data>
;
;               addr_h,addr_l are accy/sig address (same as 0xAD command)
;               CV_h, CV_l are CV address high byte first
;              data is 8 bit data for CV
;--------------------------------------------------------------------------
; 0xB0 <4 data bytes>  Reserved for future PowerCab use    (1)   0=not supported
; ***USB COMMAND ONLY***                                  !=success
;
;--------------------------------------------------------------------------
; 0xB1 xx                                           (1)   0=not supported
; ***USB COMMAND ONLY***                                  !=success
;               SET cab bus address of USB board to xx
;
;               Command Format: 0xB1 <data>
;
;--------------------------------------------------------------------------
; 0xB2 xx                                           (1)     0=not supported
;               Enable/disable echo of binary command     !=success
;               to RS-232 port.  This command is used
;               when AIU broadcast is enabled so computer
;               can distinguish between command response
;               or AIU broadcast
;
;               Command Format: 0xB2 <data>
;
;               0=no echo, 1=echo 1st byte of command, 2=echo entire command
;
;--------------------------------------------------------------------------
```

Device Warranty
This device is fully factory tested and warranted against manufacturing defects for a period of 1 year. As the circumstances under which it is installed can not be controlled, failure of the device due to installation problems can not be warranted. This includes misuse, miswiring, or operation under loads beyond the design range of the device. For warranty or non-warranty replacement send the device (and any payment, if required) to:

NCE Warranty Center
82 East Main St.
Webster, New York 14580

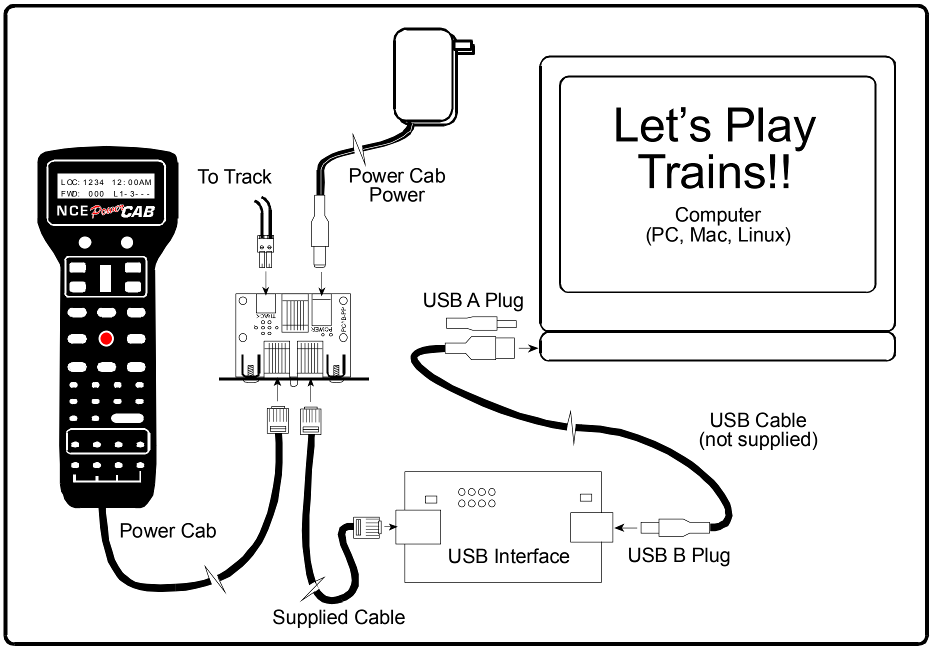
052402 23

The terms Silent Running, Power Pro, ProCab, PowerCab, the NCE logo with "The Power of DCC" slogan and the distinctive shape of the ProCab or PowerCab combined with thumbwheel and LCD are trademarks of NCE Corporation.

NOTICE: A version 1.28c software upgrade chip is included with the USB interface. This will upgrade the Power Cab for optimum performance with USB. Please install the new chip per the included installation instructions (other side of this sheet) and test the Power Cab for normal operation before using with USB.

USB -> Power Cab Quick Start:

1) Download and install Decoder Pro or other DCC software that you are going to use.
2) Download the appropriate USB driver and the driver installation manual for your computer operating system from our website: www.ncedcc.com/usb_drivers
3) Connect one end of the supplied modular cable to the USB board and the other end to the Right hand jack of the Power Cab Panel (you must remove any extra cab that may be plugged in).
4) Remove any configuration jumpers from the USB board. This will set the USB board to Power Cab 1.28c and 9600 baud.
5) Power up your computer and Power Cab.
6) Locate and connect a USB cable that will connect The USB board Type B connector to the USB connector used on your computer (usually Type A)
7) Follow the installation instructions for the computer USB driver. If you are installing on a Windows XP computer you can just let Windows search the Internet for the proper driver. It will find the right one.
8) Fire up Decoder Pro (or other software) and follow the software's instructions for initial connection and operation.



Known problems when USB board is used with the Power Cab.

Using Decoder Pro to read non-existent "indexed" CVs of QSI decoders on the program track will occasionally cause the Power Cab to crash (the LCD fills with black squares). This will require a reboot of the Power Cab and USB board. Pulling the power plug from the Power Cab Panel will cycle the power to both the Power Cab and USB board accomplishing a reboot of both item at once.