

The Sycard USBtest 2000 is a tester for USB ports. It is primarily intended for testing host controllers and hubs in a manufacturing environment. However, its precision testing capabilities and flexible software make the USBtest 2000 quite useful for engineering test and verification applications. This application note discusses the use of the USBtest 2000 in verifying hub operation where the hub under test is connected to the host controller through one or more additional hubs.

The USB revision 1.1 specification in section 7.1.19 gives the maximum number of hubs as five between the host controller and a function. In this configuration, there are six cables and five hubs that form the worst case delay and could cause the most bit erosion (impingement or closing of the signal eye pattern). The first reason to test with more than one hub in the path is for manufacturing line tests where the system under test is implemented with chained hubs. The second reason to test with chained hubs is for engineering validation or robustness testing.

As a manufacturing test, the USBtest 2000 can communicate through one or more hubs between the host controller and the hub under test. An example requirement for this type of testing is for a system where the CPU board has a hub on it to expand the number of USB ports above what available from the host controller, and where one of the CPU board hub ports connects to another hub in a monitor, docking station or keyboard. To test this second hub, the test software enumerates the hub on the CPU board and then communicates through it to the second hub. An example command is:

```
TESTHUB -m0 : 1 -m1 : 3
```

The first parameter informs the software that there is one chained hub between the host controller and the hub under test. The second parameter informs the software that the second hub is connected to port 3 of the first hub. If you also wanted to test just ports two and four of the first hub, the following command could also be run:

```
TESTHUB -p2 -z1 -z3
```

For engineering validation, several hubs can be chained together. This allows testing to the USB specified limit of 5 hubs in a chain. See the USB specification revision 1.1 figure 7-31 for an example of this configuration. The USBtest 2000 should be connected where the "Function" is shown in this diagram. The test software defaults to using downstream port number one on each hub, unless it is specified. Referring again to this figure, lets assume that the USBtest 2000 is connected to port 3 of Hub 1, hub 3 is connected to port 5 of hub 4 and all other hubs are connected using downstream port one. The following command could be used:

**TESTHUB -m0:4 -m2:5 -p1 -z1 -z2**

The first parameter specifies that four hubs are chained before the hub under test (referring to the USB spec figure, the hub under test would be Hub 1). The second parameter specifies that the second hub in the chain should use its downstream port number 5 to talk to the next hub (the second hub in the chain is identified as Hub 4 in the figure). The last three parameters specify that one port is being tested and that ports one and two are not to be tested (therefore, port 3 is the one that will be tested with the USBtest 2000).

For further engineering robustness testing, the software allows up to 10 hubs to be chained before the hub under test. This would allow a string of 11 hubs between the host controller and the USBtest 2000. While this exceeds the USB specification, in some engineering environments it may be desirable to test with a chain of more than five hubs to insure robustness.

The parameters controlling the chaining of hubs and the ports that are used are given below with the default values shown:

- 0           Select USB host controller's lowest port (port 1) (this is the default)
- 1           Select USB host controller's port 2 (not the default)
- p4          Number of hub ports to test
- zx          Hub port number to exclude from testing (default is to not exclude any)
- m0:0       Select number of additional hubs in the chain between the host controller and the hub under test
- m1:1       Selects the downstream port of the first hub to be used to continue the chain (note that the "first" hub is considered to be the hub closest to the host controller)
- m2:1       Selects the downstream port of the second hub to be used to continue the chain
- m3:1       Selects the downstream port of the third hub to be used to continue the chain
- m4:1       Selects the downstream port of the fourth chained hub
- m5:1       Selects the downstream port of the fifth chained hub
- m6:1       Selects the downstream port of the sixth chained hub
- m7:1       Selects the downstream port of the seventh chained hub
- m8:1       Selects the downstream port of the eighth chained hub
- m9:1       Selects the downstream port of the ninth chained hub
- m10:1      Selects the downstream port of the tenth hub to be used to continue the chain