

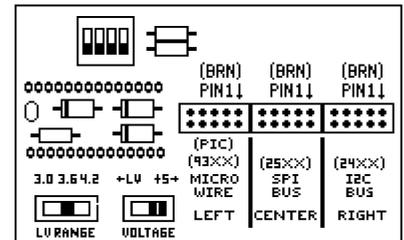
USING THE SPI SERIAL EEPROM DIRECT DRIVE CABLE ASSEMBLY (PART#CABLE8/DD)

PURPOSE

The SPI serial eeprom direct drive cable assembly is designed to support the high speed, high capacity SPI serial eeproms which find use in bios code and data storage applications on video cards, hard drives and computers. Because these parts are designed to operate at low voltages and very high speeds, reliable communication cannot be achieved using a traditional extension cable. To address this problem the DIRECT DRIVE cable assembly moves drive electronics close to the actual target memory part.

REQUIREMENTS, CONNECTIONS AND SETTINGS

The cable assembly must be used with the ASERSM1A serial eeprom adapter. This adapter provides the voltage level shifting circuitry necessary for generation of the parts low power requirements. The cable assembly must be connected to the SPI (center) position of the adapter (right figure). Note the position of the brown wire. Set the ASERSM1A adapter **VOLTAGE** switch to switch to **LV**. Set the **LV RANGE** switch to **3.0** or **3.6**. **WARNING:** This assembly is for SPI parts only. Under no circumstances should the assembly be attached to the MicroWire (93XX) or I2C (24XX) adapter positions.



OPERATION

Apply power to the programming unit, start the software and select the part with which you are working. Install the ASERSM1A adapter (fully left justified) in the programming unit socket and release the handle to lock in place. Attach the CABLE8/DD cable plug to the center (SPI) connector on the ASERSM1A. Be sure the brown wire is on the right.

DIP PARTS - If you are working with a DIP part install it in the socket on the CABLE8/DD assembly. Align the notch or dot on the chip with the notch on the 8 pin socket.

SURFACE MOUNT PARTS - If you are working with a surface mount chip attach the clip to the target chip. Align the red dot on the clip with the dot or flat side of the part. Before attempting to program the part confirm that reliable communication can indeed be achieved. To confirm reliable

access first read the part into the buffer using command 3 (Press 3 then Y). Watch the ACTIVE led on the assembly. It will light while the system reads the chip. If the ACTIVE led begins to dim or turns off before the read operation completes then the target board is drawing too much current from the programming unit. On board reading and programming will not be possible and the chip must be removed from the target board. If the ACTIVE led remains lit when the read operation completes use command 5 (buffer editor) to view the data. Be sure that there is indeed data other than all FF or 00. If it appears that the data is valid exit the buffer (press ESC) then execute command 8 (Press 8 then Y) to compare the data in the chip to the data in the buffer. Upon completion confirm that there are 000000 errors. If there are no errors then proceed with part programming. If there are errors, all FF or 00 data, then the part may need to be removed from the target board. **NOTE:** The part must be erased before programming.

ERASING THE PART - To erase the part use command Z (DEVICE OPTIONS). Press Z then E to erase the part.

The ACTIVE led will illuminate with very short blinks while the erase operation is in progress.

ADDITIONAL CONSIDERATIONS

Although the CABLE8/DD is designed to drive the part while attached to the target board, this may not be possible if the target draws more current than the programmer can deliver or interferes with the signals sent to or received from the memory part. If this occurs the part must be removed from board to permit successful reading and programming.

ACTIVE LED - The CABLE8/DD includes an led to indicate reading and programming activity. The led will light when actual signal activity occurs. If there is no signal activity or insufficient power the led will not light.

STATUS REGISTER - Many SPI memory parts include a special register as part of the memory array. This register allows areas of the memory array to be protected by setting specific bits. If you encounter a programming problem reset all bits in the status register to 0. This is performed using command Z. Press Z then R. This reads the contents of the status register. The register contents are usually a combination of 0's and 1's. Press C (change register) and use the arrow keys plus the 0 key to set any 1 bits to 0. Press ESC then press W to write the changed register image back to the part. Press ESC to exit the Z command and try programming again.

