

# USING THE PIC MICROCONTROLLER ADAPTER

The APIC1216 microcontroller adapter allows the Andromeda Research programming system to read and program all members of the 12CXXX and 16CXXX family including 8, 18, 28 and 40 pin devices. The adapter uses a 48 pin socket and a 12 position dip switch to accommodate the different device sizes (see illustrations on page 2).

## INSTALLING THE ADAPTER

Install the adapter into the 32 pin socket on the programming unit by lifting the socket handle until the pins on the base of the adapter slide into the corresponding open locations of the programming unit socket. The adapter must be fully left justified in the 32 pin socket. Once the adapter base is flush with the 32 pin socket, release the socket handle to hold the adapter in place.

## INSTALLING DEVICES IN THE ADAPTER

To install a device in the adapter, open the 48 pin socket by moving the handle to the UP position. When the device is installed, move the handle to the DOWN position to lock the device in place. Be sure the 12 position dip switch is set for the device family you are using. The control program will provide an illustration of the proper switch settings.

### DEVICE PLACEMENT (see page 2)

For 16C5X devices (illustration 1), the right side of the socket is for 18 pin devices while the left is for 28 pins. For 16C6X, 16C6XX, 7X, 7XX and 8X devices (illustration 2), the right side of the socket is for 18 pin devices while the left is for 40 pins. For 28 pin 16C6X, 16C6XX, 7X, 7XX and 8X devices (illustration 3) the devices use a portion of the socket near the center. Insert the device with the notch facing right and the left most pins aligned with the two red alignment dots. For 8 pin 12CXXX devices (illustration 4) insert the device with the notch facing right and the left most pins aligned with the two red alignment dots. **NOTE:** Devices are always inserted with the notch facing the same direction as the single red dot on the right side of the 48 pin socket.

## BUFFER DATA ORGANIZATION

The 16C5X microcontroller family are all 12 bit devices while the 16C6X, 16C6XX, 7X, 7XX and 8X are 14 bits. Data is loaded into the buffer with the lower 8 bits at even addresses and the upper 4 or 6 bits at odd addresses. Note that the upper bits of each odd address must be 0. **NOTE (16C84):** The 16C84 device has an area of 8 bit data memory. This data memory is mapped in the buffer between 800 and 87F. The 16C84 fuses are at 880.

## THE CONFIGURATION FUSES

The EPROM+ programming system provides full access to the PIC microcontroller configuration fuses. The fuses are read, changed and programmed using the "Z" - **DEVICE OPTIONS** command. When the fuses are read, the system places the value one byte past the last byte in the devices memory array.

**EXAMPLE:** The fuse byte for the 16C54 device is located at address 400H in the buffer which is one byte greater than the device size (3FFH). Fuse data placement is consistent among all members of the PIC family, regardless of which device is selected.

**FUSE PROGRAMMING** - You may program the configuration fuses using either of two techniques. The first technique is to program the fuses using the Z command. The second technique allows you to automatically program the fuses as part of the device programming process. The configuration fuses will automatically be programmed if fuse data (other than FF) is placed one byte past the last byte in the memory array. This is the same location where the data will be placed if read using the Z command. **NOTE:** If you read a PIC device using COMMAND 3 or attempt to copy a device using COMMAND 7, only the memory array will be read. The fuse data will not be read or programmed. You must use the Z command to read the configuration fuse data before it can be programmed into a device.

**NOTE:** After the system programs the configuration fuses it displays the message CONFIGURATION FUSES PROGRAMMED. This message will be displayed after the memory array has been programmed if valid fuse data exists in the buffer one byte past the last location in the memory array.

**IMPORTANT NOTE:** If you are programming a PIC device from a disk file, please be aware that the configuration fuse data may or may not be included with the file data. Due to wide inconsistencies in where the fuse data is placed in a file, the EPROM+ will only load the memory array (program) data. If the configuration fuse data is contained in the file, it is your responsibility to locate the information and place the data one byte past the last byte in the memory array.

### A NOTE ABOUT CODE PROTECTION

The entire PIC family of devices from Microchip is capable of having the memory array secured. This prevents the contents (program code) from being read in a form which allows the device to be copied. If you have a PIC device which is already programmed, it is possible that the device is secure (copy protected). You can determine this by reading the configuration fuses using the Z command. If the CODE PROTECT BIT (CP) is 0 the device is protected. There is no published method for bypassing the code protect feature.

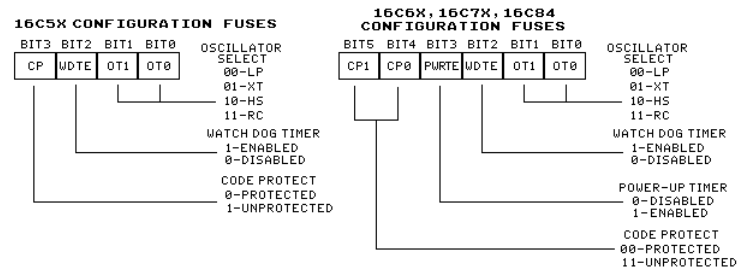


ILLUSTRATION 1 - PIC16C5X PARTS - 18 AND 28 PIN

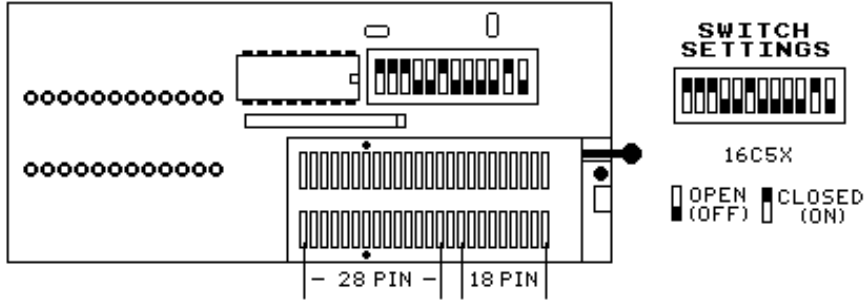


ILLUSTRATION 2 - PIC16C6X/7X/8X AND PIC16F8X PARTS - 18 AND 40 PIN

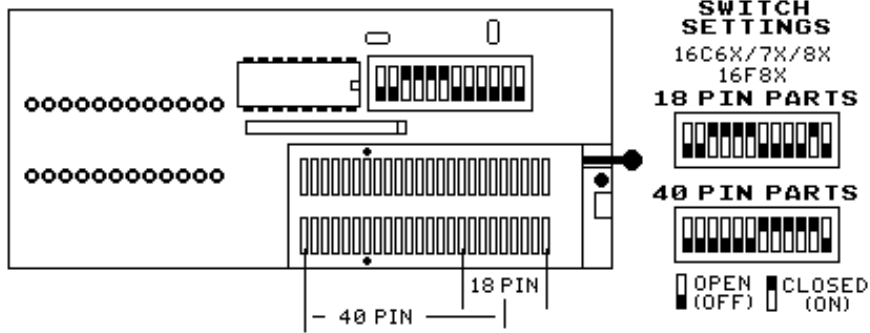


ILLUSTRATION 3 - PIC16C6X/7X/8X AND PIC16F8X PARTS - 28 PIN

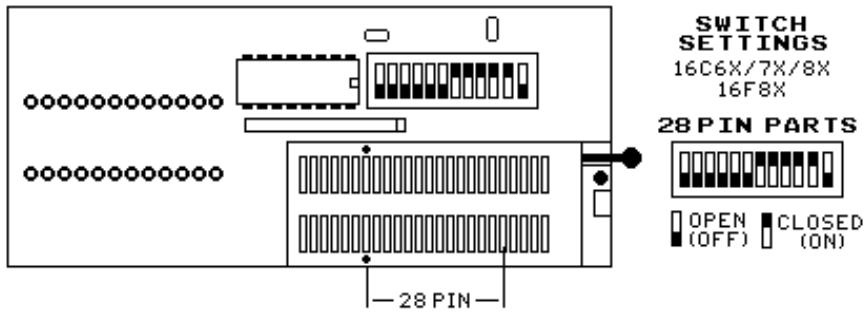


ILLUSTRATION 4 - PIC12CXXX PARTS - 8 PIN

