

Memory Map for the MCU320 board:

The Intel 8051 MCUs and all derivatives are based on the Harvard architecture. This is to say that they have separate memory space for program (CODE) and external data (XDATA). The advantage to this is two fold. The address space is doubled and memory management is simplified in simple systems. The disadvantage is that no writes are allowed to code space. This was not a problem when PROM was used for program memory, but with the advent of FLASH ROM, the micro can benefit from the option to write to this nonvolatile memory. In addition, FLASH ROM is less expensive due to its packaging. To accommodate this and other features the MCU320 board supports three modes of addressing.

MODE 0:

Harvard architecture (Intel mode). The board has 8k of SRAM (or NVSRAM) and up to 64k of program memory in FLASH ROM. Although the program memory is flash, no write cycles are allowed to this space in this mode. Off board peripherals should be limited if possible to the top 8k of XDATA. The mid 48k of XDATA can be von Neumann (common code and data space) mapped on a monitor card for debugging so this space should not be used if the associated monitor card is used for the debug cycle. If one chooses to use a ROM emulator, this limitation is not applicable.

XDATA MEMORY MAP		CODE MEMORY MAP	
0xFFFF	PERIPHERALS	0xFFFF	FLASH ROM 2
0xE000			
0xDFFF			
	RESERVED		FLASH ROM 1
0x2000	MCU SRAM	0x8000	
0x1FFF		0x7FFF	
0x0000		0x0000	

MODE 1:

von Neumann architecture (FLASH write mode). In this mode all peripherals are disabled by disabling the RD and WR lines on the back plane. All but the lower 8k of program memory is then von Neumann mapped to allow write cycles to this space. This mode is specifically used for the storage of data in non-volatile FLASH program memory. This mode can be optionally controlled by the MCU through any P1 pin. It is possible to use this mode in a stand alone boot-loader mode. It should be noted that if a boot-loader is put into the lower 8k of program memory, a jumper can be set on the board to allow in system firmware updates. If one of the P1 pins is used, the firmware may be updated over a network (CAN) through a modem or through any of the UARTs including the one on the monitor board.

XDATA MEMORY MAP		CODE MEMORY MAP	
0xFFFF	FLASH ROM 2	0xFFFF	FLASH ROM 2
0x8000		0x8000	
0x7FFF	FLASH ROM 1	0x7FFF	FLASH ROM 1
0x2000			
0x1FFF	MCU SRAM		
0x0000		0x0000	

MODE 2:

Monitor mode. This mode is used to develop and debug programs. The idea is to use a monitor board to plug into the back plane. This board disables the MCU board based CODE memory and supplies 8k of monitor ROM mapped to CODE space. The monitor card also supplies SRAM von Neumann mapped to the mid 48k of both CODE and XDATA space. The RAM on the MCU board is still available in the lower 8k of XDATA. Programs are downloaded through a high speed off chip UART so that the timers and UARTs are available for in circuit debug. The monitor firmware allows the code to then be debugged. The MCU has no direct way of controlling this mode.

XDATA MEMORY MAP	
0xFFFF	PERIPHERALS
0xE000	
0xDFFF	
	CODE SRAM
0x2000	
0x1FFF	MCU SRAM
0x0000	

CODE MEMORY MAP	
0xFFFF	NOT USED
0xE000	
0xDFFF	
	CODE SRAM
0x2000	
0x1FFF	MONITOR ROM
0x0000	

Peripheral Space:

This is the space used to interface peripherals. This includes the UARTs, network controllers, digital i/o, analog i/o etc. Unless the peripheral requirements exceed the allowed 8k of address space, devices should not be mapped into the space below.

0xFFFF	SELECT 7
0xFC00	
0xFBFF	SELECT 6
0xF800	
0xF7FF	SELECT 5
0xF400	
0xF3FF	SELECT 4
0xF000	
0xEFFF	SELECT 3
0xEC00	
0xEBFF	SELECT 2
0xE800	
0xE7FF	SELECT 1
0xE400	
0xE3FF	SELECT 0
0xE000	

Eight select lines are available on the MCU board. These select lines are used to identify individual boards with common peripheral (i.e. Two identical UART boards can be distinguished by jumper setting the select for each board). This allows up to eight 110 volt input boards to co-populate a bus with up to eight analog output boards. In order for this scheme to work, a fixed address offset is allied to the base of the selected slot by each unique board. For instance, a UART board is mapped to the last sixteen bytes of the select space. Therefore since the size of any select space is 1k or 0x03FF bytes, the offset applied by the UART card is 0x03F0 or 0x03FF-0x000F. If SELECT 0 is used, the addresses used by a UART board are 0xE3F0 to 0xE3FF. If SELECT 7 is used as on the Monitor board, the addresses used are 0xFFFF0 to 0xFFFFF.

The following is a table showing the memory allocation to date. This will change as more i/o peripherals are developed.

ADDRESSES WITHIN SELECTED SPACE	BOARD DESCRIPTION
0x03F0 to 0x03FF	UART, USART, DUART, MUART etc.
0x03EE to 0x03EF	16 channel isolated input/output