

Quick Start To Commodore 64 Machine Language Programming

The quickest way to start putting machine-language instructions into a Commodore 64 is by simply POKEing them in. The POKE command on a C64 takes the following format:

```
POKE address,value
```

For example, the following command:

```
POKE 100,20
```

...Would insert the value 20 into memory address 100.

The following command will place a capital letter A in the middle of the screen:

```
POKE 1483,1
```

Suppose we want to make a short program in machine language to do this. What we need to do is take a certain value and store it in a specific memory location, and have a program to do this for us.

The program would then be in two parts. First, it would need to load the value to be stored into the CPU's accumulator. The Commodore 64 uses the 6510, which is pin- and instruction-compatible with the 6502, so we can use 6502 opcodes to do this stuff. In the 6502's case, the instruction to load the accumulator (assembler mnemonic LDA) has a machine language opcode of A9 hexadecimal, which is 169 decimal. So the first two bytes of our program are as follows:

```
A9 01
```

(The second byte is the value of 1, which is the C64's code for a capital letter A, and the value we want to load into the accumulator.)

After this, we need to store the accumulator in memory location 1483 decimal, which is 05CB hexadecimal. The opcode for store accumulator (STA) is 8D hex (141 decimal). The STA instruction takes an odd form in memory: After the 8D opcode, the next byte needs to be the LOW byte of the destination address, and the byte after that is the HIGH byte of the destination address. We want to store the accumulator in memory location 05CB, so to represent this as a machine language instruction, we'd need to first put the instruction opcode (8D), the low byte of the address (CB), and finally the high byte of the address (05). Our fully-rendered machine language program to print an A on the screen, then, is:

A9 01 8D CB 05

That's all very well and good, but how do we get this program into memory and running? By POKEing it, of course. Suppose we want to insert and run this program from memory location 4000 hexadecimal (16384 decimal). You can run a machine-language routine from the C64's memory by simply using the SYS command. For example, if you type the following command:

```
SYS 16384
```

...That will start a machine-language program at memory location 4000 hexadecimal (16384 decimal).

Knowing this, we can now begin POKEing our machine-language program into memory. There's just one catch you need to know: When you're using the SYS command, the machine code routine must end with an RTS opcode to return program control to BASIC when the routine is done. The RTS opcode is 60 hexadecimal (96 decimal), so add that to the end of the program above. The program we'll use, then, is:

```
A9 01 8D CB 05 60
```

The following commands will POKE this program into memory at 4000h.

```
POKE 16384,169 ;LDA
POKE 16385,1   ;1 (code for letter A)
POKE 16386,141 ;STA
POKE 16387,203 ;low byte
POKE 16388,5   ;high byte
POKE 16389,96  ;RTS
```

When you've typed all this, you can finish off by typing SYS 16384 and the program should work. Congratulations! You've written, programmed, and run a machine-language program on a Commodore 64!