**AT89S51/52 ISP Programmer – Erase**

By [Himanshu Choudhary](https://www.engineersgarage.com/author/hchoudhary/)

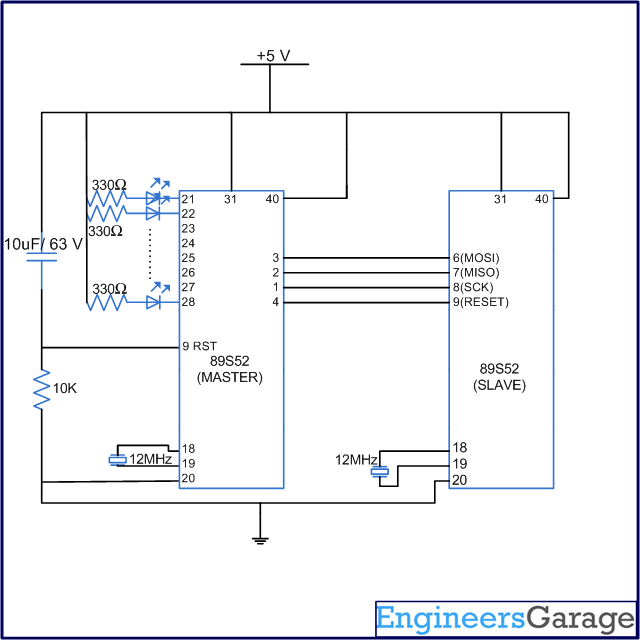
[Facebook](https://www.engineersgarage.com/#facebook)[Twitter](https://www.engineersgarage.com/#twitter)[LinkedIn](https://www.engineersgarage.com/#linkedin)[Reddit](https://www.engineersgarage.com/#reddit)[Pinterest](https://www.engineersgarage.com/#pinterest)

[Share](https://www.addtoany.com/share#url=https%3A%2F%2Fwww.engineersgarage.com%2Fat89s51-52-isp-programmer-erase%2F&title=AT89S51%2F52%20ISP%20Programmer%20%E2%80%93%20Erase)

This is the first operation that needs to be performed on the target [microcontroller](https://www.engineersgarage.com/microcontroller). It is assumed that the target microcontroller has some code in its flash memory. If it does not has any content, write any program in the memory using a programmer. Whenever the flash memory of the chip is erased successfully, it shows ‘**FF**‘ in all its memory locations. In this project we will erase an 89S51 microcontroller which has some preloaded program in its memory from another microcontroller.

The reader should know the basics of sending and receiving the single byte in programming mode. Refer to 8051 Programmer basics before reading this tutorial.

The flash memory of the microcontroller is electrically erasable, *i.e.*, it needs electrical pulses to erase the memory. When the four byte instruction corresponding to erase is given to the target microcontroller in the programming mode it erases its internal flash memory. It is important to erase the memory every time the microcontroller is programmed. Once the instruction corresponding to erase has been received the entire memory gets erased, i.e., it does not happens that if the program stored in the memory is occupying only 10% of the entire memory space only this much memory will be erased and rest will be not be erased. In order to perform any operation during the programming mode, the first instruction should always be Programming Enable. It needs to be sent only once and then any operation can be carried out.

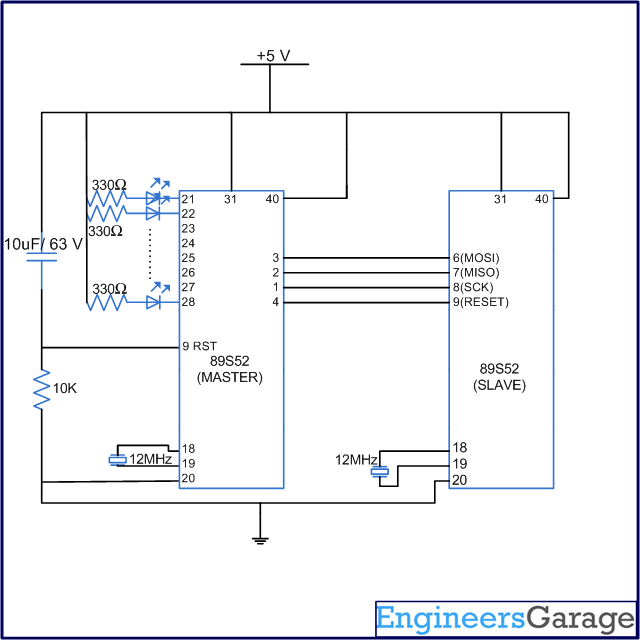


*Fig. 1: Table listing 8051 instructions for programming enable and chip erase*

The above table shows the instructions corresponding to programming enable and chip erase. After sending the program enable instruction the target microcontroller sends a byte with value 0x69, which means the programming enable mode has been successfully enabled.

**Circuit Diagram and Algorithm**

**Circuit Diagram**



*Fig. 2: Circuit Diagram of 8051 Programmer*

**Algorithm**

The algorithm for erasing the chip is as follows:

1.      Power on the circuit.

2.      Send the instruction corresponding to programming enable.

3.      Check the 4th byte on MISO pin. If we receive 0x69 this means chip is enabled.

4.      If programming mode in enabled then send four bytes instruction corresponding to chip erase.

**Code**

**Code**

#include<REG51.h>

#include<intrins.h>

#define port P1

sbit sck=port^0;

sbit res=port^3;

sbit miso=port^1;

sbit mosi=port^2;

bit bit1;

unsigned int i,bitno=0;

unsigned char a,b,c,d;

void delay(unsigned int msec) // Function for delay

{

int i,j;

for(i=0;i<msec;i++)

for(j=0;j<1275;j++);

}

void sendbit()

{

mosi=bit1;

delay(1);

\_nop\_();

sck=1;

delay(1);

\_nop\_();

sck=0;

}

void sendbyte(unsigned char m)

{

for(bitno=0;bitno<8;bitno++)

{

bit1=m/128;

m=m<<1;

sendbit();

}

}

void main()

{

i=0;

sck=0;

res=1;

delay(500);

//=======Program enable===========

a=0xac;

b=0x53;

c=0x00;

d=0x11;

sendbyte(a);

sendbyte(b);

sendbyte(c);

sendbyte(d);

//==========Erase==================

a=0xac;

b=0x82;

c=0x12;

d=0x13;

sendbyte(a);

sendbyte(b);

sendbyte(c);

sendbyte(d);

res=1;

while(1);

}