



# MICROCONTROLLER PROJECT LABORATORY

EDUCATIONAL STUDIES PROGRAM – HIGH SCHOOL STUDIES PROGRAM – SUMMER 2001  
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

## PROGRAMMING THE P89C51RD2

This handout describes the hardware and three software methods for programming the microcontroller.

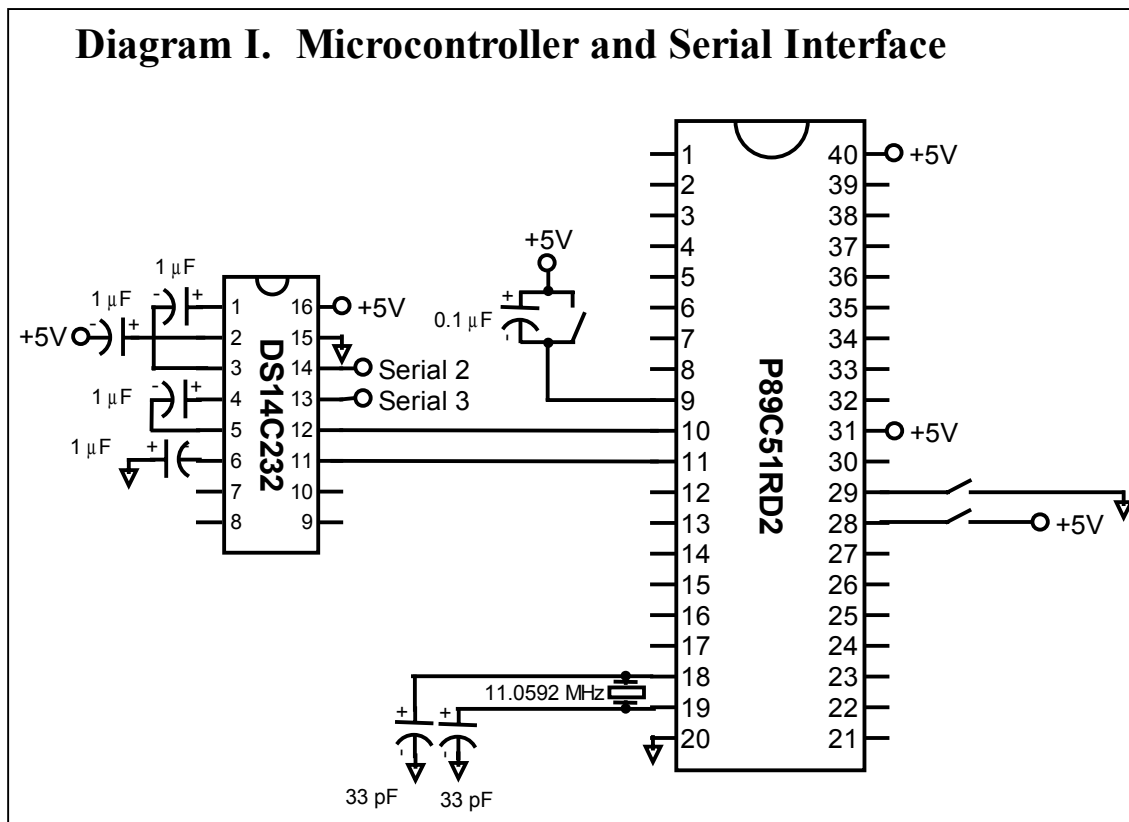
### 1 HARDWARE

Diagram I shows the hardware configuration needed for programming. The requirements are:  
Microcontroller:

1. Pins 40 and 31 connect to +5V.
2. Pin 20 connects to ground.
3. Capacitor and switch connect pin 9 to +5 V.
4. Switches connect pin 29 to ground, and pin 28 to +5 V.
5. 11.0592 MHz crystal connects pins 18 and 19.
6. Pins 18 and 19 each connect to ground via 33 pF capacitors.

Serial level converter:

1. Pin 16 connects to +5 V.
2. Pin 15 connects to ground.
3. 1  $\mu$ F capacitors connect 1+ to 3-; 2+ to +5 (-); 4+ to 5-; ground+ to 6-.
4. Pins 12 and 11 connect to microcontroller pins 10 and 11.
5. Pins 14 and 13 connect to serial port pins 2 and 3.



The most basic way to program the microcontroller is through plain serial communication. This method requires no special software.

### 2.1 Microsoft Windows

Open the Hyperterminal program (under Accessories, Communication). Create a new connection, and pick an available port, and configure it for 19200 baud, 8 data bits, no parity, 1 stop bit, and no flow control. You can save the connection and make a shortcut to it on your desktop to avoid repeating this setup step.

### 2.2 Hardware

Hold down the right switch (pin 29 to ground, pin 28 to +5 V), and click the left switch (pin 9 to +5V). If you don't have a working switch, you can connect and disconnect wires. This should reboot the microcontroller into programming mode. You may need to hit reset more than once if it doesn't work.

### 2.3 Initialization ( Required before erasing or programming )

Enter a capital U. You should see the U echo back on your screen. This indicates that you are talking to the microcontroller's programming software. This echo is required in order to proceed. If you don't get a "U" see section 5, Troubleshooting, at the end of this handout.

Then set the timer for 11 MHz by pasting the following command into the terminal window:

```
:010000020BF2
```

(See ISPCOMMANDS.TXT for a complete list of commands)

### 2.4 Erasing

If there is old code stored on the microcontroller, you must erase it before programming new code. Send this command to erase block zero:

```
:020000030100FA
```

### 2.5 Programming

To program code onto the chip, paste the .OBJ file into the terminal window. You can either open the .OBJ file in a text editor and then manually copy and paste, or use the Hyperterminal command called "Send Text File" under the Transfer menu that will do it for you.

Important: Watch as the data scrolls by. Each successive line should have the 5<sup>th</sup> digit increase by one. If the 5<sup>th</sup> digit skips, some data has been lost and you can try again.

### 2.6 Running Code

After programming code, you can run it by clicking the reset button without the right button connected.

Before the first time you ever run code, you must set the Status Byte to zero with these commands:

1. Erases the status byte and boot vector: **:020000030400F7**
2. Set the boot vector to FC: **:030000030601FCF7**
3. Set the status byte to zero: **:03000003060000F4**

This procedure only needs to be performed once per chip.

WinISP is a program that lets you push buttons rather than pasting binary commands.

#### 3.1 Initialization:

1. Select P89C51RD2 for Chip.
2. Select a port.
3. Enter 11 for Osc (MHz).
4. Click the middle “Read” button (not “Read Part”).  
The response should be “Boot Vector Read Okay”

#### 3.2 Erasing

1. Click the “Erase Blocks” button.
2. Pick the top block, 0.
3. Hit the “ERASE!” button.

#### 3.3 Programming

1. Click the “Load File” button.
2. Select the .OBJ file you want to program.  
(You will need to type the name or “\*” because the filter is for .HEX files only by default.)
3. Click the “Program Part” button.  
The response should be “Flash Programming Successful.”
4. You can optionally choose “Verify part” to make sure there was no error, but I have never found errors when programming with WinISP.

#### 3.4 Running Code

Before the first time you ever run code on a chip, the Status Byte must be zero.

If the Status byte is not zero after you click “read”, then you should change it to zero and hit “Write”.

RD2MON is a program that runs on the P89C51RD2 to let you upload code.

- Compared to just Hyperterminal, it has a nicer user interface (no pasting binary commands).
- Compared to WinISP, it lets you see serial information from programs in the same window that you used to program them, and you don't have to quit and restart WinISP.

#### 4.1 Installation

To install RD2MON, first program its code using either Hyperterminal or WinISP.

**IMPORTANT: Be sure RD2MON is uploaded correctly before proceeding:**

In WinISP, hit "Verify"

From Hyperterminal, send the following command and look for any gaps of all "FFFFF"

**:05000004700075000012** [hit enter]. Upload it again if you're not sure.

Next, set the Boot Vector to 70. (That causes it to execute the program at address 7000 when you reboot in programming mode.) From WinISP, enter "70" in the Vector box, and hit write.

From plain hyperterminal, send the following commands:

- ( Do initialization if you haven't already: **U :010000020BF2** )
1. Erase Status Byte and Boot Vector: **:20000030400F7**
  2. Set Status Byte to Zero: **:03000003060000F4**
  3. Set Boot Vector to 70: **:0300000306017083**

#### 4.2 Using RD2MON

When you reboot into programming mode, you should get the message:

```
Welcome to Microcontroller Project Laboratory 2001!  
Commands: [E]rase [D]ownload [B]oot Vector [G]oto#### [R]ead#### [W]rite####=#  
  
RD2MON>
```

Now you get the simple way of doing things:

To program, first hit E0 to erase block zero.  
Then hit D to download, and paste the object file (using "Send Text File" in Hyperterminal).

To run a program you just uploaded, type "G000" to execute code at location 0.

Read and Write let you examine and modify the flash memory.  
Don't modify memory beyond 7000 or you may corrupt RD2MON.

To uninstall RD2MON, hit B for boot vector and type y, then FC.

If you can't get a "U" back from the serial port:

Check your wiring:

- 1) Make sure the blue and purple wires from the serial port connector aren't reversed. When plugged in to a computer, a voltmeter should give at least +/- 8 volts on each wire.
- 2) Check your switches. Make sure the switch on the right connects pin 29 to ground and 28 to +5. Use a voltmeter to check. Use wires if your switch doesn't work.

Check your computer's serial port settings:

- 1) 19200 baud, N-8-1, no flow control.
- 2) An obscure possibility: make sure you have FIFO enabled and set at maximum.  
In Windows NT: Start → Settings → Control Panel → Ports → Settings → Advanced  
In Windows 2000: Start → Settings → System → Device Manager → Ports → COM# → Port Settings → Advanced

Try a patch:

Try also holding pin 27 to +5 V, or disconnecting it.

Check your chip:

Bring your chip to a TA to have them test it in a working system. This will determine whether your chip is disabled or whether your wiring or computer is at fault.

Your chip could be disabled if the Boot Vector is set improperly, or if it is physically damaged. In either case, you can get a new chip.