

# ISPV3 Programmer's Guide

## 1 Introduction

This guide addresses the features, setup and operation of the CRD89C51xxx microcontrollers with ISPV3 firmware.

The firmware is intended to provide In-system / In-application programming interface for the CRD89C51RD, CRD89L51RD, CRD89C512RD and CRD89C51RE devices through the UART interface.

The benefits of the ISPV3 firmware include:

- The firmware only takes 1K of Flash memory
- Same firmware for all devices
- Short instruction set with few ASCII conversions
- Supports Data Flash programming and read-back

## 2 ISP timeout period

The ISP timeout period refers to the delays after a device reset where the ISP firmware will wait from the serial port. If during the timeout period no valid ISP command is received the ISP firmware will jump to the user program. If there is a successful command sent, the ISP will not time-out again until the device is reset.

The default ISPV3 WDT time out period depends on the device on which the ISP firmware is installed:

- 0.84 sec @ 40MHz (CRD89C51RD), 131ms (CRD89C512RD)
- 1.52 sec @ 22MHz (CRD89C51RD), 131ms (CRD89C512RD)
- 2.8 sec @ 12MHz (CRD89C51RD), 131ms (CRD89C512RD)

Shorter ISP start-up time out periods for the CRD89C51RD should be selectable from software. The watchdog timer period of the CRD89C512RD is independent of the  $F_{OSC}$  frequency, but is operated from a 250 kHz oscillator coupled to a pre-scaler and a 16-bit timer. The longest possible timeout period for the CRD89C512RD is 262 ms.

The CRD89L51RD does not include a watchdog timer on chip. Timer 2 overflow is used instead. Flash address 0xFBFE contains the Timer 2 overflow period (default is 0xFF), when the default timeout is defined as 10 x Timer 2 (16 bit) overflow

The Timer 2 overflow is calculated as:

- 0.196 sec @ 40MHz
- 0.350 sec @ 22.11MHz
- 0.65 sec @ 12 MHz

The firmware does not differentiate which device it is running on so Timer2 or WDT will define the actual Timeout period of the ISP firmware. The "default" Timeout value chosen is the best compromise found to accommodate all devices. The Versa ware ISP software provides specific configuration options for each specific device.

## **2.1 Rescue Feature of the ISPV3 firmware:**

Because neither the firmware nor software can know which device is installed it is possible to set a timeout value so that the software will not connect with the device anymore. In that case, the firmware has a provision built in to allow re-synchronization with the ISPV3 Firmware.

Keeping pin P4.2 Low at device reset will force the maximum timeout period possible with the firmware, which will be:

- CRD89C512RD:
  - 262ms
- CRD89C51RD:
  - 1.67 sec @ 40MHz
  - 3.03 sec @ 22.11MHz
  - 5.59 sec @ 12 MHz
- CRD89L51RD:
  - 5.03 sec @ 40MHz
  - 9.10 sec @ 22.11MHz
  - 16.7 sec @ 12 MHz

## **3 Resetting the device**

The firmware only starts at device power on, meaning that for any given configuration, the device is not likely to be ready to receive commands. Versa Ware ISP uses the RTS line of the RS232 port to send a reset pulse on the development board, but any unused line can be used in custom designs.

Reset the device after programming in order to run the loaded program.

## **4 Command Set**

### **4.1 Terminology**

- All commands and responses are presented in tables
- Unless otherwise specified, each cell represents one byte (8 bits)
- Bytes are transmitted in order from left to right
- Any multi-byte value is transmitted high order byte first, unless otherwise specified
- Numbers in cells represent the value to be sent
- Single characters are ASCII values to be sent
- In the case of an ASCII character that is a number, it will be followed in its cell by its ASCII code

### **4.2 Checksum**

Any command sent or response received that has a checksum as part of its structure uses the formula below:

Checksum = (Sum of all bytes in packet) MOD 256

All checksums are one byte long. If the firmware gets a command with an incorrect checksum, it will return 3 bytes:

?	S	146
---	---	-----

### 4.3 Command structure

All commands follow the same structure except for synchronization. The general command is given below:

Star	Size	Command	Argument	Zero	Checksum
------	------	---------	----------	------	----------

**Star:** The star character (ASCII 42)

**Size:** The number of bytes after this entry

**Command:** A one letter ASCII code. A table of valid commands is available in the next section

**Argument:** If the command requires arguments, they are placed here

**Zero:** The value 0

**Checksum:** The checksum for this packet

### 4.4 Command list

The following table lists all of the commands available for ISPV3:

Name	ASCII	Value
Connect	x	120
Erase	E	69
Page Erase	S	83
Program	P	80
Read	R	82
Set zone	Z	90
Write	W	87

If a command is sent that does not appear on this list, the firmware will respond:

?	C	130
---	---	-----

If a command is not sent in time, the firmware will respond:

?	T	147
---	---	-----

#### 4.4.1 Connect

This command is used to start communicating with the firmware. This allows for the automatic baud rate algorithm to attempt to synchronize to the host software. To use, simply send a lowercase x on the UART:

X
---

If the firmware can synchronize, it will reply:

Y	3 (ASCII 51)
---	--------------

If any other value is returned, it may be because of the following:

- Wrong port. Try connecting on another port.
- Wrong speed. Try connecting at a different speed.
- Wrong chip. This may be a device that does not have ISPV3 firmware.

This table shows the most likely synchronization speeds for various baud rates.

f (MHz)	115200	57600	38400	19200	9600	4800	2400
40	X						
22.18	X						
20		X	X				
16.384		X	X				
16			X	X			
14.746	X	X	X				
12							
11.059	X						
10			X				
8				X	X		
7.373	X	X	X				
4					X	X	

If a crystal frequency has more than one possible connection speed, there is the possibility that a connection will fail at one speed, but work at another and vice versa. It is critical to try to connect at all possible speeds for a given crystal before failing a device.

#### 4.4.2 Erase

This command will set all non-ISP flash addresses to 0xFF. The firmware will verify whether all areas are indeed 0xFF. This may take up to 2 seconds.

To use, send this string:

*	3	E	0	114
---	---	---	---	-----

If the erase is successful, the firmware will respond:

E	0	69
---	---	----

Otherwise it will respond:

E	!	102
---	---	-----

If an error is returned, the device may have flash reliability issues, which can be caused by too many erase and programming operations on the same byte of flash (over 100000).

### 4.4.3 Page Erase

This command, formerly known as Sector Erase, will erase 1 page of flash and verify that it has been erased (all locations 0xFF). A page is 512 Bytes.

To use, send this string:

*	4	S	Page	0	Checksum
---	---	---	------	---	----------

Page is a 1-Byte value specifying the page number. If the erase is successful, the firmware will respond:

S	0	83
---	---	----

Otherwise it will respond:

S	R	165
---	---	-----

A page erase can be refused for 2 reasons:

- The page is in the ISP zone
- The security settings do not allow page erasing

### 4.4.4 Program

Use this command to load a string of bytes into the flash. This is its command structure:

*	Size	P	Address	Data Size	Bytes to load	0	Checksum
---	------	---	---------	-----------	---------------	---	----------

**Address:** The 16-bit address (2 Bytes)

**Data Size:** The number of bytes to load (1 byte)

**Bytes to load:** The values to place in the flash (size of the value of the Data Size parameter)

The first byte will be placed at Address, the next at Address + 1, and so on. While this structure allows up to 249 values to be sent, it is not recommended to send more than 32. This is because any transmission error will take longer to be relayed, and the longer the string, the longer and less predictable the timeout period the host software will have to have.

The firmware will verify if each byte is written correctly, so there is no need to read back the flash to compare buffers.

If the program was successful, the following is returned:

P	0	80
---	---	----

If there was a programming error, the following is returned:

P	!	113
---	---	-----

If the command was refused, the following is returned:

P	R	162
---	---	-----

A program command can be refused for if security settings do not allow programming.

#### 4.4.5 Read

Use this command to read back bytes from any address in the flash. This is the command structure, where Address is a 16-bit (2 Byte) value:

*	5	R	Address	0	Checksum
---	---	---	---------	---	----------

If the command is successful, the byte is returned in this structure:

R	Byte	Checksum
---	------	----------

Otherwise it will respond:

!	!	66
---	---	----

A read can be refused if the security settings do not allow page erasing.

#### 4.4.6 Set Zone

This command is used to set the flash zone. The CRD89C512RD has 2 flash zones: a program flash, where the executable programs and the ISP reside, and a data flash, where non-executable code can be stored. The Read, Write, Program, Erase, and Page Erase commands work on the zone selected by this command, and do so until the next Set Zone command is received. After a reset, the zone is set to point to the program flash.

Setting the flash zone on other devices has no effect and all commands will work properly, regardless of the zone setting. This makes it impossible to determine the device by switching zones and checking for an error condition. However, if a byte is written to the data flash, and the same address is read from the program flash, and the 2 values are different, then the device is definitely a CRD89C512RD.

To send a Set Zone command, use this structure:

*	4	Z	Zone	0	Checksum
---	---	---	------	---	----------

Zone is a value of either 0 (program flash) or 1 (data flash).

If the command is successful, the following is returned:

Z	Zone	Checksum
---	------	----------

Otherwise,

Z	!	123
---	---	-----

This error code is sent if a value other than 0 or 1 is sent as the Zone.

### 4.4.7 Write

Use this command to set one byte of data at a particular address. Here is its command structure:

*	6	W	Address	Byte	0	Checksum
---	---	---	---------	------	---	----------

Address is the 16-bit address, and Byte is the value to write. If the write is successful, the following is returned:

W	0	87
---	---	----

If there was an error while writing, this string is returned:

W	!	120
---	---	-----

If the write was refused, this string is returned:

W	R	169
---	---	-----

A write can be refused if the security settings do not allow page erasing.

## 5 Flash program memory structure

<b>ISPV3 Firmware Area</b>	0xFFFF
<b>1024 Bytes (1 KB)</b>	
ISPV3 Flash Configuration 5 Bytes	0xFC00 0xFBFF 0xFBFB 0xFBFA
<b>User Program Area</b>	
61899 Bytes (~ 60.5 KB)	
Interrupt Vector Area 47 Bytes	0x0030 0x002F 0x0001
<b>ISP Mode Configuration (1 Byte)</b>	0x0000

## 5.1 ISPV3 Firmware Area (0xFC00 – 0xFFFF)

This is the where the firmware resides. Erase, Page Erase, Program, or Write commands will not modify this area of the flash. This area can be read by the Read command if security settings allow it.

## 5.2 ISP Mode Configuration (0x0000)

**Address 0x0000 must be set to 0xFF. Failure to do so will result in the device running in normal mode and the ISPV2 firmware will not be executed.** Addresses 0x0000–0x0002 are typically used to place an LJMP statement to the main block of code by compilers. Since 0x0000 must be 0xFF, the main start address should be placed in the flash configuration area.

### 5.2.1 Retrieving the jump vector from a file

By default all 8051 compilers put instruction LJMP (0x02xxxx) at address 0x0000 in the Hex File. It is often undesirable or impossible to modify source code to leave 0xFF at address 0x0000 in order for the ISP to start. Therefore programming software used to communicate with the ISP firmware should take care of this and do the following:

- In the file, identify the line containing the data to be written from address 0x0000. Usually it is the first line of an Intel Hex File.
- Grab the 2 bytes address following the LJMP instruction and do not write anything at address 0x0000 or write 0xFF at address 0x0000 instead.
- Write the 2 bytes address in flash as follows:
- Write the MSB of the LJMP address (location 0x0001) at 0xF1FD.
- Write LSB of LJMP address (location 0x0002) at 0xF1FC
- Write all other data contained in the hex file line (address >= 0x0003) to their normal destination.
- Address 0x0003 and next contains interrupt vector jumps, so they must be written “as is” in flash
- Write all other Hex File data normally

This C code is what Versa Ware ISP uses to extract the LJMP instruction:

```
//////// Get jump address
//
// Written for Ramtron International Corporation
// 8/03/2004
//
// This function searches a file and gets the value of the
// jump address from the 0000 line
//
// Returns non-negative value for the address
//
////////////////////////////////////

#include <stdio.h>
#include <string.h>

int getJump(char* hx)
{
    const char SEARCH[] = "00000002";
```



```
char line[1024];

// If the file doesn't have code at 0x0000
// Give a value that is not possible

int found = -1;
char address[5];
address[4] = '\0'; // End in NULL

FILE* hex = fopen(hx, "r"); // Open the text file to read

// Return if there is no file
if(!hex) return found;

// Search the whole file
while(fgets(line, sizeof(line), hex))
{
    if(!strncmp(line + 3, SEARCH, 8))
    {
        // An address starts at the 11th character in an Intel
        // Hex File
        strncpy(address, line + 11, 4);
        sscanf(address, "%x", &found);
        break;
    }
}

fclose(hex); // Close files after using
return found;
}
```

## 5.2.2 Firmware recovery

In case any value other than 0xFF is placed at address 0x0000, any parallel programmer can be used to restore the ISPV2 firmware. **No physical damage is done to the device.**

## 5.3 Interrupt Vector Area (0x0001-0x002F)

This is where interrupt vectors must be placed. See the device datasheets for more information. Normally, vectors are 3 bytes long aligned at addresses of multiples of three. The first vector starts at 0x0003. (Addresses 0x0000–0x0002 are typically used to place an LJMP statement to the main block of code by compilers. Since 0x0000 must be 0xFF, addresses 0x0001 and 0x0002 can be used for other data.)

## 5.4 User program area (0x0030-0xFBFA)

Make sure that user programs are placed in this range.

## 5.5 ISPV3 Flash Configuration (0xFBFB-0xFBFF)

Address	Value	Default
0xFBFF	Security Monitor Configuration	OFF
0xFBFE	Timer 2 timer period	See section 1
0xFBFD	MSB of user program start address	0x01
0xFBFC	LSB of user program start address	0x00
0xFBFB	Watchdog Timer period	See section 1

### 5.5.1 Security Monitor Configuration

7	6	5	4	3	2	1	0
X	X	G	G	R	R	P	P

- G:** Set the bit low (0) to prevent flash page erase
- R:** Set the bit low (0) to prevent flash reading
- P:** Set the bit low (0) to prevent flash programming
- X:** These bits are reserved, leave them high (1)

Both bits must be set to 0 for the protection to take effect.

**Warning:** Preventing flash reading will prevent the access of this byte. Security must be set after all relevant operations are performed.

### 5.5.2 User program Start Address (0xFBFD:0xFBFC)

Place the start address of the user program here. The firmware will use the value stored at this location to start the user program. If no value is written here, (0xFFFF is placed here when erasing) the firmware will jump to 0x0100.