

Reference Guide For SRecCvt

An S-Record Formatting Program
For
The M68HC12 Microcontroller Family

Written By
Gordon Doughman
Software Specialist

1.0 Introduction

The Motorola S-Record object file format was developed in the early days of microcontroller development as a method of representing absolute binary object code in an ASCII text format. The original format consisted of three S-Record types that defined a header record (S0), a program/data record (S1) and logical end-of-file record (S9). The S1 record contains a 16-bit load address and S9 record contains a 16-bit program start address. With the introduction of the M68000 family in 1979 several additional S-Record types were defined to accommodate its 32-bit linear address space. The S2 and S3 S-Records are similar to S1 records except that they have a 24-bit and 32-bit address field, respectively. Likewise, S7 and S8 S-Records are similar to the S9 record except that S7 records contain a 32-bit program start address and S8 records contain a 24-bit program start address. Because all Motorola microprocessors and microcontrollers developed until that time utilized a linear address space, the address fields used in each of the S-Records reflected a simple linear address.

Because the M68HC12 family is the first Motorola microcontroller to utilize a paged program memory architecture, the linear address paradigm of the S-Record standard does not allow a simple encoding of paged program addresses. Unfortunately, because an S-Record standard was not established for the paged memory architecture of M68HC12 family members containing more than 64K bytes of Flash program memory, two different approaches have been used by various software tool vendors.

One of these formats uses the S2 record in a nonstandard manner to encode the PPAGE number in the upper 8-bits of the 24-bit address while using the lower 16-bits of the S2 address field for the PPAGE window address. While this scheme might be adequate for programming tools supplied by the vendors supporting this format, S-Records formatted in this manner would not be properly interpreted by general device programmers. In addition, such S-Records are not accepted by Motorola for the generation of masked ROM devices. This S2 record format is known as the 'banked' S-Record format.

The second format employed by software tool vendors utilize S-Records in a standard manner supplying only a simple linear address for all code and data. For M68HC12 devices having more than 64K bytes of address space, the program memory address space is viewed as a simple linear array of memory that begins at \$00000. This is the same format in which S-Records need to be

presented to a stand alone nonvolatile memory device programmer or sent to Motorola for the generation of a masked ROM device. For example, a device such as the MC68HC912DA128 that contains 128K of on-chip flash memory would have S-Record addresses from \$00000 through \$1FFFF. The mapping of the linear address contained in the S-Record to the 16K byte page view able through the window at addresses \$8000 through \$BFFF is the responsibility of the S-Record loader and shown in Figure 1.

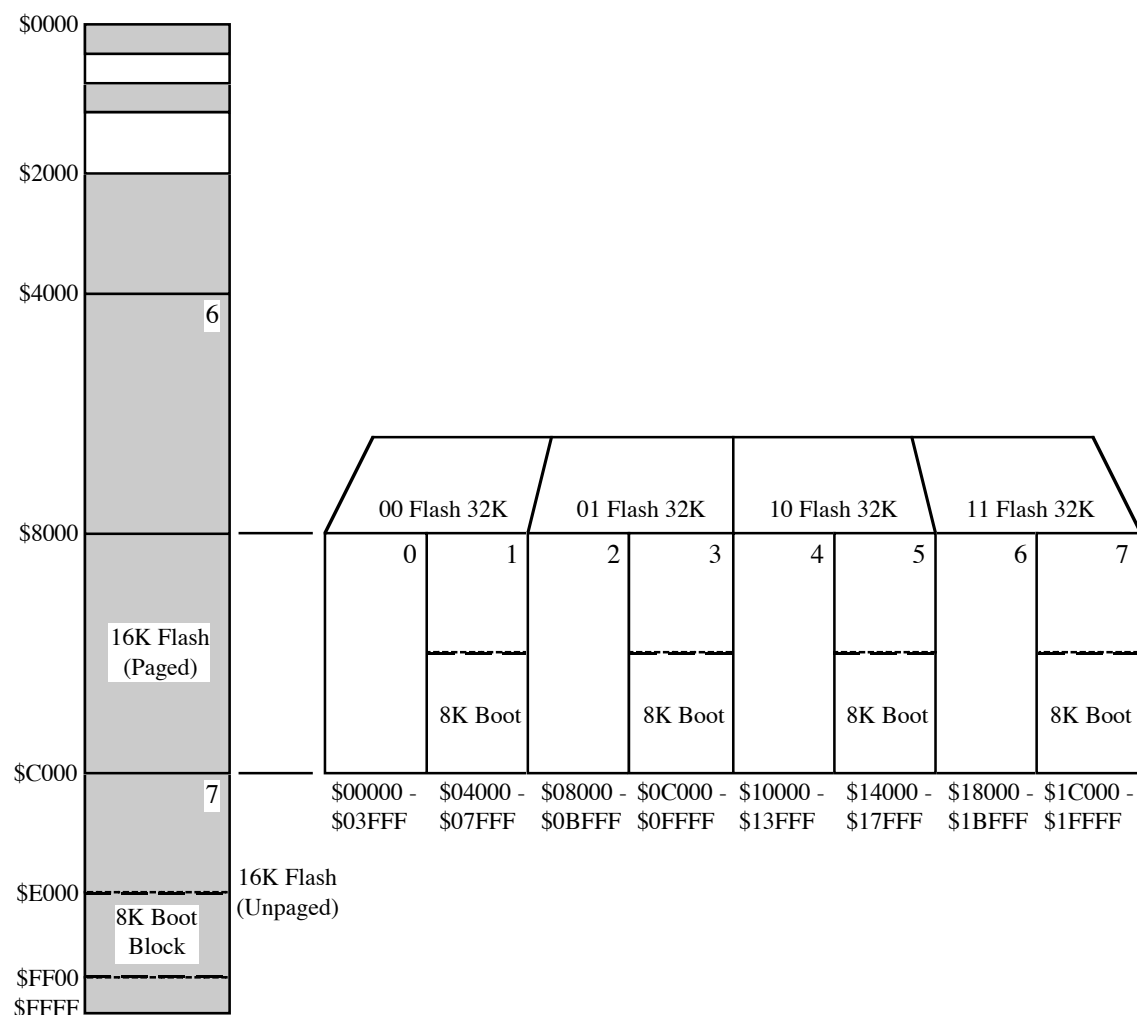


Figure 1, MC68HC912DA/DG128 Flash Memory Paging

The `SRecCvt` program was originally developed to help customers bridge the gap by providing a way to convert the 'banked' S2 S-Record format to the linear address format required by Motorola's D-Bug12 and the P & E Microcomputer's Prog12S Flash programming tools.

2.0 SST 0.5μ Flash Programming Requirements

Recent developments have resulted in the licensing of small sector Flash memory technology from SST by Motorola. This Flash technology, utilizing a split gate, has many advantages over the Flash technology used in the original M68HC912 family of devices (known as UDR or 1.5T Flash). However, the programming requirements of the 0.5μ SST Flash are significantly different from the UDR Flash. While the UDR Flash can be programmed a single byte or aligned word at

a time, it is recommended that the 0.5 μ SST Flash be programmed a 64 byte aligned block at a time (the block address must begin on a 64 byte boundary).

Because S-Records may start at any address, may contain data between one and 251 bytes and can appear in any address order in the file, the S-Records generated by assemblers and/or linkers are likely to be unsuitable for programming M68HC912 devices with 0.5 μ SST Flash using tools such as D-Bug12 or Prog12S. The SRecCvt program provides options to reformat S-Record files so that data is contained in 64 byte S-Records, all beginning on a 64 byte boundary. In addition, several options are available for producing properly formatted S-Record files that can be submitted to Motorola for masked ROM generation.

3.0 2T Flash Programming Requirements

The two transistor Flash technology originally developed for the M68HC08 family has been integrated into the M68HC12 family. This Flash technology can be easily merged with Motorola's Hyper-integration process, allowing high voltage, high current I/O on the same die as the CPU. While the UDR 1.5T Flash can be programmed a single byte or aligned word at a time, the 2T Flash should be programmed an 8 byte aligned block at a time (the block address must begin on an 8 byte boundary).

Because S-Records may start at any address, may contain data between one and 251 bytes and can appear in any address order in the file, the S-Records generated by assemblers and/or linkers are likely to be unsuitable for programming M68HC912 devices with 2T Flash using tools such as D-Bug12 or Prog12S. The SRecCvt program provides options to reformat S-Record files so that data contained in S-Records all begin on an 8 byte boundary and have a length that is a multiple of 8 bytes.

4.0 Using the SRecCvt Program

The SRecCvt program is provided as a DOS command line application and as a Macintosh standalone application. Under Windows 95 or 98, SRecCvt is invoked from a DOS command line window. On a Macintosh, the program is invoked by double clicking the application icon and entering the command options in the dialog box that appears. Typing the command followed by a carriage return will display the command's usage as shown in Figure 2.

To convert an S-Record file containing banked S2 S-Records to the linear address S-Record format, simply enter the command name (SRecCvt) followed by the appropriate command line options and the path and file name of the file to be converted. An output file named Out.S19 will be created containing the object file in the linear address S-Record format. For example, the command line

```
sreccvt -b 16 0x30 -o LinearOutFile.S19 BankedInFile.S19
```

will convert the banked S-Record file, BankedInFile.S19, generated for the MC68HC912DP256 to linear S-Records placing the result in LinearOutFile.S19. Note that the '-o' command line option is used to supply an output file name other than the default.

Note: Some Linkers and/or S-Record generation utilities that produce the banked S2 S-Records may include EEPROM data in the same S-Record file as the Flash memory data. When converting banked S2 S-Records to the linear address S-Record format make sure that the input file does **NOT** contain S-Records outside the range of the on-chip Flash memory.

```
SRecCvt v1.0.25
```

```
Usage: SRecCvt [options] Path/FileName
```

```
Options:
```

```
-s2  Output only S2 records
-b  <NumPages> <PPBase> [<SRecBlkSize>]
-pw <PageWinStart> <PageWinSize>
-ob  Output banked S-Records
-o  <Path/OutputFileName>
-dt128a  Convert input to 64 byte S-Records
-d60a  Convert input to 64 byte S-Records
-ga32  Convert input to 8 byte boundary S-Records
-m  <MapStartAddr> <MapEndAddr> <SRecBlkSize>
-gmb  <GlobalMapBias>
-es  <FlashErasedStateByte>
-f  <FillByte>
-rf  Fill Memory Map With Random Data
-of  <Offset>  Add offset to each S-Record Load Address
-lp  Convert linear $4000 - $ffff to paged S-Records
-crc  Calculate a 32-bit CRC over the memory range
-cl  <Path/LinearFileName> Compare Linear SRec File to Input File
-cb  <Path/BankedFileName> Compare Banked SRec File to Input File
```

Figure 2, SRecCvt Usage Display

5.0 Options

With some exceptions, the options may be entered in any order on the command line and may be any combination of upper and lower case characters; case is not significant. In addition the input path and/or file name does not have to appear as the last item on the command line.

5.1 -s2 Option

The **-s2** option can be used when converting banked S2 S-Records or reformatting linear address S-Records. When used, the output file will contain all S2 records even if the input file contains a mixture of S1 and S2 records. This option must be used on S-Record files when submitting them to Motorola for masked ROM generation.

5.2 -b Option

The **-b** option is used to specify the banked memory characteristics of parts containing more than 64K of Flash memory. The **<NumPages>** parameter is used to specify the number of 16K byte banked Flash pages contained on the part. The **<PPBase>** parameter specifies the PPAGE value that selects the lowest numbered 16K page visible through the address window at \$8000 - \$BFFF. The third parameter, which is optional, may be used to specify the length of the generated S-Records. If the **-b** option is not present on the command line the default value used for the **<NumPages>** parameter is 8 and the default value used for the **<PPBase>** parameter is 0x00. The default S-Record size is 32 bytes. The default values are correct for M68HC12 128K parts. The **<NumPages>** and **<SRecBlkSize>** parameters must be **decimal** numbers and the

<PPBase> parameter must be a **hexadecimal** number.

Note: The **-b**, option may be used when converting banked S-Records to the linear S-Record format or, if the **-ob** option is used, when reformatting banked S-Records.

Example:

The MC68HC912DP256 has a 1MB address space and contains 256K of on-chip Flash memory. The on-chip Flash memory is addressed as the highest 256K block of the 1MB address space, requiring a PPAGE value of \$30 to access the first 16K page through the PPAGE memory Window. To convert a banked S-Record file generated for the MC68HC912DP256 to linear S-Records, the following command line could be used:

```
srecvt -b 16 0x30 -o MyOut.S19 InFile.S19
```

Notice that the optional <SRecBlkSize> parameter was not provided as part of the **-b** command line option. This will cause the default S-Record size of 32 bytes to be used.

5.3 -pw Option

The formulas shown in Figure 3 are used when converting S-Record load addresses between Linear and Banked/Paged formats. The default values used by SRecCvt for PAGEWinStart and PAGEWinSize are \$8000 and \$4000 respectively. This corresponds to the start address and size of the Flash PPAGE window on the S12 and S12X microcontrollers.

```
PageNum = LinearLoadAddr / PAGEWinSize;  
  
PAGEWinAddr = (LinearLoadAddr % PAGEWinSize) + PAGEWinStart;  
  
LinearLoadAddr = (PageNum * PAGEWinSize) + (PAGEWinAddr - PAGEWinStart);
```

Figure 3, S-Record Load Address Conversion Formulas

The **-pw** option is provided primarily for use when converting S12X banked EEPROM S-Records where the PAGEWinStart and PAGEWinSize are \$800 and \$400 respectively. When converting to or from S12X banked EEPROM S-Records, the **-pw** option **must** be used in conjunction with the **-b** or **-ob** options to specify the Page Window start address and the Page Window size. The <PageWinStart> and <PageWinSize> parameters must be **hexadecimal** numbers.

Example:

The MC9S12XDP512 contains 4K bytes of EEPROM. One kilobyte of the EEPROM is accessible at \$0C00 - \$0FFF in the CPU12X's 64K address space. The remainder is accessible through the 1K byte EPAGE window at \$0800 - \$0BFF. The EEPROM is also accessible in the upper 4K of the 256K of Global Address space allocated for EEPROM at \$13F000 - \$13FFFF. The following command line can be used to convert a banked EEPROM S-Record file to an S-Record file with Global load addresses:

```
srecclvt -b 4 0xfc -pw 800 400 -gmb 100000 -o MyOut.S19 BankedEEInFile.S19
```

Note: See the description of the `-gmb` option for additional details.

5.4 -ob Option

The `-ob` option is used to override the default S-Record file format generated by the SRecCvt program. When SRecCvt is used to convert banked S-Records, the output file would usually contain linear S-Records. However, if the `-ob` option is present on the command line, the output file will contain reformatted banked S-Records. While this option may seem to defeat the purpose of the SRecCvt program, it actually has a valid use. Use of this option allows banked S-Records to simply be reformatted rather than converted. Reformatting banked S-Records is useful for development tools, such as MCUEz, that support direct loading of banked S-Records.

Example:

As mentioned in Section 2.0, M68HC12 family devices that use the 0.5 μ SST Flash technology require that the Flash be programmed a 64 byte aligned block at a time. Because the S-Record file generated by MCUEz may not meet these requirements, SRecCvt may be used to reformat such an S-Record file. The following command line could be used to reformat a banked S-Record file generated for an MC68HC912DT128A:

```
srecclvt -b 8 0x00 64 -ob -o MyOut.S19 InFile.S19
```

The `-ob` option may not only be used for reformatting banked S-Records, but it can also be used to convert linear S-Record files to banked S-Record files. The command line for converting a linear S-Record file generated for an MC68HC912DT128A to a banked S-Record file would be:

```
srecclvt -dt128a -ob -o MyOut.S19 InFile.S19
```

5.5 -o Option

The `-o` Option is used to specify an output file name other than the default, `Out.S19`. The supplied path and/or file name must conform to the rules defined by the host operating system and should be no longer than 255 characters.

5.6 -dt128a Option

The `-dt128a` option is one of four options that causes the `SRecCvt` command to reformat linear S-Records rather than convert banked S2 S-Records. The option is actually an abbreviated form of the `-m` option that specifies the `<MapStartAddr>` as 0x00000, the `<MapEndAddr>` as 0x1FFFF and the `<SRecBlkSize>` as 64. Because the S-Record input file is converted from ASCII to binary data and placed in a memory array, there is no restriction on the address order or length of the input S-Records.

To ensure the smallest possible output file size, the memory array is filled with the erased state (0xFF) of the Flash memory before the conversion process is begun. The output file will only contain S-Records for those 64 byte memory areas that contain one or more data bytes other than 0xFF. Note that the `-f` option may be used to override this behavior by filling the memory array with a value other than 0xFF.

5.7 -d60a Option

The `-d60a` option is one of four options that causes the `SRecCvt` command to reformat linear S-Records rather than convert banked S2 S-Records. The option is actually an abbreviated form of the `-m` option that specifies the `<MapStartAddr>` as 0x1000, the `<MapEndAddr>` as 0xFFFF and the `<SRecBlkSize>` as 64. Because the S-Record input file is converted from ASCII to binary data and placed in a memory array, there is no restriction on the address order or length of the input S-Records.

To ensure the smallest possible output file size, the memory array is filled with the erased state (0xFF) of the Flash memory before the conversion process is begun. The output file will only contain S-Records for those 64 byte memory areas containing one or more data bytes other than 0xFF. Note that the `-f` option may be used to override this behavior by filling the memory array with a value other than 0xFF.

5.8 -ga32 Option

The `-ga32` option is one of four options that causes the `SRecCvt` command to reformat linear S-Records rather than convert banked S2 S-Records. The option is actually an abbreviated form of the `-m` option that specifies the `<MapStartAddr>` as 0x8000, the `<MapEndAddr>` as 0xFFFF and the `<SRecBlkSize>` as 16. Because the S-Record input file is converted from ASCII to binary data and placed in a memory array, there is no restriction on the address order or length of the input S-Records.

To ensure the smallest possible output file size, the memory array is filled with the erased state (0x00) of the 2T Flash memory before the conversion process is begun. The output file will only contain S-Records for those 16 byte memory areas containing one or more data bytes other than 0x00. Note that the `-f` and `-es` option may be used to override this behavior by filling the memory array with a value other than 0x00.

5.9 -m Option

The `-m` option is a generalized form of the S-Record reformatting specifier. It is provided to support M68HC12 or other MCU devices that may require a memory map and/or block size other than those defined by the `-dt128a`, `-d60a` or `-ga32` options. The size of the memory map supported by `SRecCvt` is limited only by the amount of memory available on the host computer system. The `<MapStartAddr>` and `<MapEndAddr>` must be hexadecimal numbers and can optionally be preceded by the characters '0x'. `<MapStartAddr>` must be less than

<MapEndAddr>. The size of the memory map as defined by <MapEndAddr> – <MapStartAddr> + 1 must be evenly divisible by the specified S-Record block size. The <SRecBlkSize> specifier must be a **decimal** number and can be a minimum of eight and a maximum of 251. Because the S-Record input file is converted from ASCII to binary data and placed in a memory array, there is no restriction on the address order or length of the input S-Records.

To ensure the smallest possible output file size, the memory array is filled with the erased state (0xFF) of the Flash memory before the conversion process is begun. The output file will only contain S-Records for those memory blocks (defined by the <SRecBlkSize> parameter) containing one or more data bytes other than 0xFF. Note that the -f and -es options may be used to override this behavior by filling the memory array with a value other than 0xFF.

Example:

To generate a linear S-Record test file for the MC68HC912DP256, the following command line could be used:

```
sreccvt -m c0000 fffff 32 -rf -o MyOut.S19
```

Note that because the -rf option was used, an input file name was not required.

5.10 -gmb Option

The S12X microcontrollers unify the on and off chip memory resources and I/O registers in to a single 8M byte Global memory space. Within this 8M byte memory space, Flash, EEPROM, RAM and I/O registers are allocated fixed amounts of the address space as shown in Figure 4.

Memory Type	Address Range	Size
I/O Registers	\$000000 - \$0007FF	2048 Bytes
RAM	\$000800 - \$0FFFFFFF	1MB - 2K
EEPROM	\$100000 - \$13FFFF	256KB
Off Chip Memory	\$140000 - \$3FFFFFFF	2.75MB
Flash	\$400000 - \$7FFFFFFF	4MB

Figure 4, S12X Global Resource Memory Map

Examining the S-Record load address conversion formulas in Figure 3, it can be seen that for Flash memory, conversion from banked to linear load addresses would produce addresses in the range of \$000000 - \$3FFFFFFF. To produce S-Records having Global load addresses for Flash, an additional Global Map Bias must be added to the calculated linear load address to form a Global load address. Note that the same issue exists for the EEPROM when converting from banked to global load addresses. In the EEPROM case, conversion from banked to linear load addresses would produce addresses in the range of \$000000 - \$03FFFF.

The -gmb option allows for the specification of this additional conversion factor. When converting S-Records from banked to global or vice versa. When converting Flash S-Records for the S12X, a bias of \$400000 should be used. When converting EEPROM S-Records for the S12X, a bias of \$100000 should be used.

Note: When simply reformatting S-Records with banked or Global load address, the -gmb option should **not** be used. The term reformatting refers to when the input and output files contain S-Records with the same type of load addresses.

Example:

The MC9S12XDP512 contains 4K bytes of EEPROM. One kilobyte of the EEPROM is accessible at \$0C00 - \$0FFF in the CPU12X's 64K address space. The remainder is accessible through the 1K byte EPAGE window at \$0800 - \$0BFF. The EEPROM is also accessible in the upper 4K of the 256K of Global Address space allocated for EEPROM at \$13F000 - \$13FFFF. The following command line can be used to convert a banked EEPROM S-Record file to an S-Record file with Global load addresses:

```
srecsvt -b 4 0xfc -pw 800 400 -gmb 100000 -o MyOut.S19 BankedEEInFile.S19
```

Note: See the description of the -pw, option for additional details.

5.11 -es Option

The -es option is used to specify the value of an erased byte of Flash memory. This option is primarily intended to be used with the -m option allowing complete flexibility when specifying a memory map other than one of the the default maps. The default value of an erased byte of Flash memory is 0xFF.

Note: The -dt128a, -d60a, -ga32 or -m option must appear on the command line before the -es option because the -es option cannot be used when converting banked S2 S-Records to linear address S-Records.

5.12 -f Option

To ensure the smallest possible output file size when reformatting a linear S-Record file, the memory array is filled with the erased state of the Flash memory before the process is begun. Normally, the output file will only contain S-Records for those memory blocks containing one or more data bytes other than the erased state of the Flash. The -f option is used to override this default behavior by filling the memory array with a value other than the erased state of the Flash memory. The <FillByte> parameter must be a hexadecimal number, can optionally be preceded by the characters '0x' and must have a value 0x00 through 0xFF. This option is useful for filling unused memory areas of an MCU with a single byte instruction such as SWI.

Note: The `-dt128a`, `-d60a`, `-ga32` or `-m` option must appear on the command line before the `-f` option because the `-f` option cannot be used when converting banked S2 S-Records to linear address S-Records.

5.13 -rf Option

The `-rf` option is not really related to the process of reformatting a linear S-Record file. Instead it produces an S-Record output file where the S-Records contain 'random' data. Many times programming test files will contain data that matches or is somehow related to the lower byte of the address of the programmed memory location. Generating an S-Record file containing 'random' data is useful for testing the Flash memory arrays on an MCU where it may be suspected that a problem exists relating to the value of the lower byte of the address. The `-rf` option does not require an input file name. If one is supplied, it is simply ignored. The `-o` option may be used to specify an output file name.

Note: The `-rf` option CAN be used to generate test files containing banked S-Records.

Example:

The following command line will generate a banked S-Record file for the MC68HC912DP256 with each S-Record containing 'random' data.

```
srecvt -b 16 0x30 32 -rf -o MyOut.S19
```

5.14 -of Option

The `-of` option allows a fixed offset to be added to the load address of each S-Record as it is decoded and data is placed in an internal memory array.

Note: The `-dt128a`, `-d60a`, `-ga32` or `-m` option must appear on the command line before the `-of` option because the `-of` option cannot be used when converting banked S2 S-Records to linear address S-Records.

5.15 -lp Option

There are cases where a developer needs the peripherals of an MCU containing more than 64K bytes of Flash memory, but has modest code space requirements. In such cases the developer might want to treat the 48K of Flash memory visible from \$4000 - \$FFFF as a simple linear address space rather than utilize the program memory paging system. The `-lp` option can be used to convert S1 S-Records in the \$4000 - \$FFFF range to properly formatted S-Records for programming an M68HC12 paged memory device.

Example:

The following command line will convert S-Records in the \$4000 - \$FFFF address range to

linear S-Records that are properly formatted for programming an MC9S12DP256 with tools requiring linear S-Records..

```
sreccvt -m c0000 fffff 32 -lp -o <outfilename> <inputfilename>
```

The following command line will convert S-Records in the \$4000 - \$FFFF address range to linear S-Records that are properly formatted for programming an MC68HC912DA128 with tools requiring linear S-Records.

```
sreccvt -m 0 1ffff 32 -lp -o <outfilename> <inputfilename>
```

5.16 -crc Option

The `-crc` option is used to calculate a 32-bit CRC over the address range of the specified memory map containing the data from the S-Record file `<inputfilename>`. Before placing the the S-Record file data in the memory map, the memory map is filled with the erased state of the flash or the value specified by the `-es` option. The generating 32-bit polynomial used for the CRC calculation is 0x04c11db7, which is the same polynomial used for the ethernet CRC calculation. The initial CRC value used by the calculation is 0xffffffff and the ones compliment of the calculated CRC value is displayed.

Example:

The following command line will calculate a 32-bit CRC value for a linear S-Record file formatted for programming an MC9S12DP256.

```
sreccvt -m c0000 fffff 32 -crc <inputfilename>
```

5.17 -cl and -cb Option

The `-cl` and `-cb` options are used to compare the contents of a linear or banked S-Record file, respectively, with the contents of the specified input file. The data contained in each S-Record file is read into a memory array as specified by the `-b` or `-m` options, and then compared a byte at a time. Before the comparison process is begun, each memory array is filled with the erased state of the Flash memory (default value of 0xFF or as specified by the `-f` option) before the comparison process is begun. Any differences between the contents of the two S-Record files is reported in the output file.

Note that by combining the `-b` and `-m` options with the `-cl` and `-cb` options, a linear address S-Record file can be compared to a linear address S-Record file, a banked address S-Record file can be compared to a linear address S-Record file or a banked address S-Record file can be compared to a banked address S-Record file. For example, the following command line:

```
sreccvt -m c0000 fffff 32 -cl D:\S12Dx256.S19 -o D:\Diffs.txt Test.S19
```

compares the contents of the linear S-Record file `S12Dx256.S19` to the contents of of the linear S-Record file `Test.S19`. and report the differences, if any, in the file `Diffs.txt`. The following command line compares two banked S-Record files:

```
sreccvt -b 16 30 32 -cb D:\S12Dx256Banked.S19 -o D:\Diffs.txt TestBanked.S19
```

Finally, the following command line compares a banked S-Record file to a linear S-Record file:

```
srecvct -b 16 30 32 -cl D:\S12Dx256.S19 -o D:\Diffs.txt TestBanked.S19
```

The differences file has the following format:

File 1: D:\S12Dx256.S19

File 2: D:\Test.S19

File 1 Address	File 1 Data	File 2 Address	File 2 Data
\$000C03E0	\$46	\$000C03E0	\$56
\$000C8780	\$30	\$000C8780	\$70

As this example shows, two memory locations were found that did not match. The contents of each memory location along with its address is displayed. Note that the `-ob` command line option can be used with the `-cl` or `-cb` options to display these addresses in banked format. The example below displays the addresses shown in banked form.

File 1: D:\S12Dx256.S19

File 2: D:\Test.S19

File 1 Address	File 1 Data	File 2 Address	File 2 Data
\$30:83E0	\$46	\$30:83E0	\$56
\$32:8780	\$30	\$32:8780	\$70