

# Notes on Sxxx Commands in Macros

The SRS controller commands beginning with an S take arbitrary strings as parameters. The parameter string is terminated with a carriage return.

An Sxxx command (S130) is used to define the contents of a macro. This results in an inability to create a macro with an embedded Sxxx command. The carriage return terminator for the string parameter of the embedded command is recognized by the S130 set macro command as its terminator, and is therefore not embedded in the macro. For example, to send the string, “abcde” out serial port 2, you would use the command:

```
S377abcde<CR>
```

To set macro 23 to send the string, “abcde” out serial port 2, you need to embed that command sequence into macro 23. The expected command sequence to do this would be:

```
S13023 S377abcde<CR><CR>
```

This will not work. The controller command processor will recognize the first carriage return as the end of the parameter for the S130 set macro command. The result is that macro 23 will be loaded with:

```
S377abcde
```

Instead of:

```
S377abcde<CR>
```

When the macro is executed, it will not find a terminator for the “abcde” string, and will not send anything. In addition, these strings would display identically in the configuration program macro editor, so the cause of the problem would not be obvious to a site administrator.

CPU code version 0126 attempts to solve this by translating a \ character to a carriage return on execution of any Sxxx command. This is not entirely satisfactory. Change macro commands cannot be embedded in a macro, and the \ is lost on reading the macro into the configurator program.

CPU code versions 0127 and higher solve this problem by providing an alternative method for delineating the parameter string of the Sxxx commands. As an alternative to the <CR> terminator, the string can be surrounded by a matching pair of square brackets, [ and ]. Any bracket pair not starting with the first character of the string is ignored and handled as any other character by the command. This enables nested brackets and macros which modify other macros. Special handling for the \ character has been removed.

For example, the controller will send “abcde” out serial port 2 if either of the following two command sequences is sent to the controller:

S377abcde<CR>

S377 [abcde]

Note that S377 [abcde], with a space between the S377 and the [ will not work, as the [ is not the first character. If there were a following carriage return, as in S377 [abcde] <CR>, then “[abcde]” would be sent out the serial port.

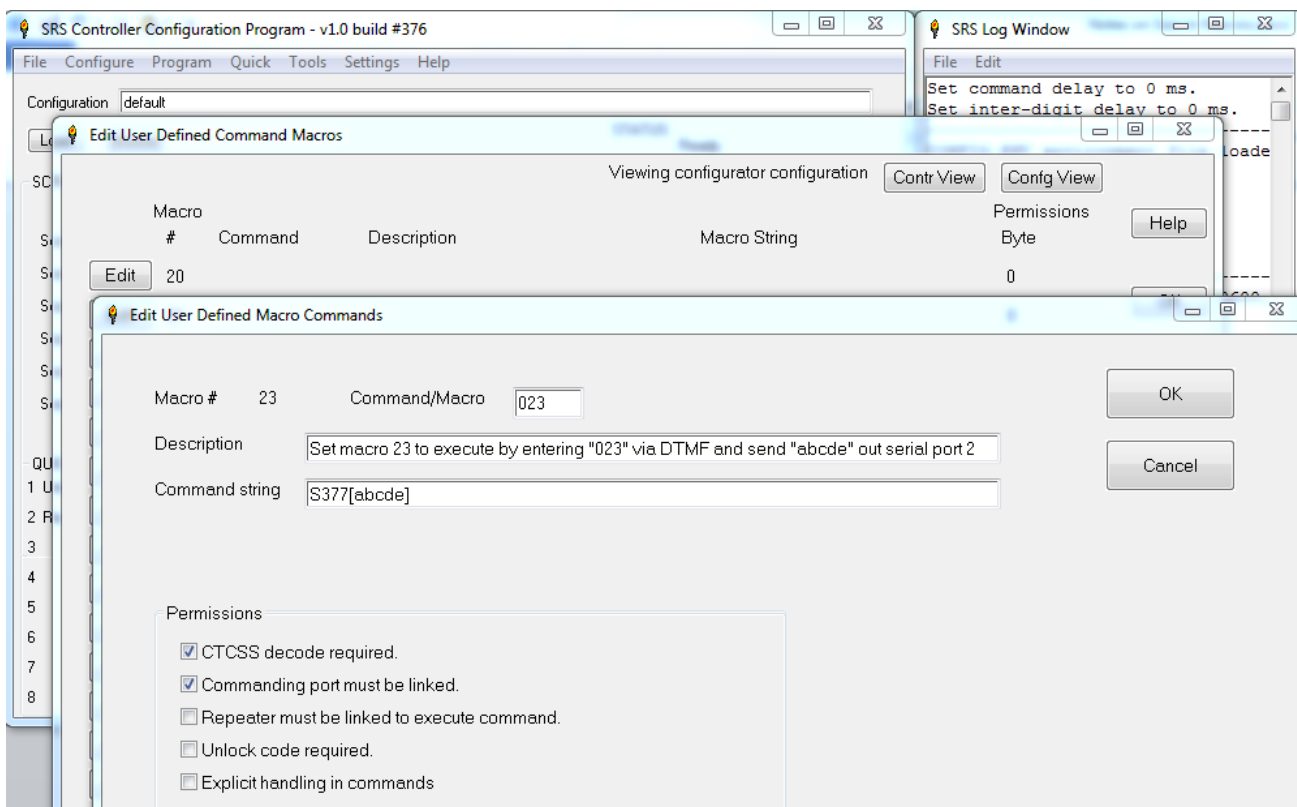
To set macro 23 to send the string out the serial port, the command sequence to the controller would be either one of the following:

S130 [23 S377 [abcde] ]

S13023 S377 [abcde] <CR>

The configurator program uses the S130 command with a terminating <CR> to define the macros. So, to set macro 23 to send abcde to serial port 2, the correct string to put in the macro editor window is:

S377 [abcde]



The configurator program will generate and send the second S130 sequence above when commanded to send the macros to the controller.

With this approach, it is even possible to create macros which change other macros. For example, to define another macro (say macro 24 for this example) to change macro 23 to send "fghi" to serial port 2, you would enter the following into the configurator's definition window for macro 24:

```
S130[24 S377[fghi]]
```