Micro-Robotics Ltd
Cambridge CB5 8QB
+44 (0) 1223 523100
www.microrobotics.co.uk

Venom-SC Tutorial Guide: Update

This document is intended to provide an update to *Venom-SC: A Tutorial Guide* as the language has been improved since the *Tutorial* was printed in 2003.

This document does not cover the new objects added to the system. Instead you are referred to the *Venom-SC Help File* where every object type is thoroughly documented.

The *Venom-SC Help File* may be downloaded from our website, or accessed from within the *VenomIDE* development environment.

A new Reference

All references in the *Tutorial* to *Venom-SC Object/Language Reference* should be changed to refer to the *Venom-SC Help File* instead. The following pages have one or more of these references:

Pages 4, 33, 51, 69, 75, 78, 81, 85, 86, 100, 104, 111, 116, 130, 132, 135, 136, 142, 148.

Updates to the text

The following are updates to the text in the *Tutorial*, listed under existing headings where possible, and with page numbers to help you find the original text.

If you are using the paperback edition of the *Tutorial* then it may be useful to go through it and annotate the margins in pencil to indicate those places where entries are now expanded.

GETTING STARTED page 6

There has been considerable improvement in the development tools since the *Tutorial* was written.

To get hold of the latest tools you should

1. Download and install **VenomIDE**, our Integrated Development Environment from our website at www.microrobotics.co.uk
2. Go to the menu **Help > IDE contents**, from where you will see a link to the New User Tutorial. This replaces the entire first chapter of the *Tutorial*.

Note: VenomIDE will not work on Mac or other non-windows operating systems. If you are using one of these then please use the Getting Started chapter as-is.

YOUR DEVELOPMENT ENVIRONMENT page 46

This whole chapter is now redundant except for those users who can't use *VenomIDE*, as mentioned in *Getting Started*.

DEBUGGING

Finding procedure lines page 66

Finding where in your code an error originated is much easier using *VenomIDE* – see *Getting Started*, above, for how to get hold of this.

To find the line at which an error occurred, you only have to double-click the error listing on the terminal to be taken to the correct part of the source code. This applies to both Syntax and Runtime errors.

Please see the *VenomIDE* help system for more details – i.e. NOT the language help file, but the one for the *VenomIDE* application. You can start this by hitting F1 when *VenomIDE* is open.

VARIABLES AND EXPRESSIONS

Arithmetic operators page 22

There is now a 'Power of' operator: X ^ Y.

Please see the *Venom-SC Help File* for full details.

FURTHER EXPRESSIONS page 75

There are now more escape sequences in addition to the ones listed in the *Tutorial*: \a \b \f \n \r \t \v.

Look up 'escape' in the *Venom-SC Help File* index for more details.

FURTHER OBJECTS

Creating Temporary Objects page 84

The keyword AUTODESTRUCT may now be used to automatically remove temporary objects that were created for use within the lifetime of a particular procedure. Its operation is detailed in the *Venom-SC Help File*.

There is also now a 'Garbage scanner' – this is an operating system tool that may be used to find 'memory leaks' – i.e. badly written programs that don't remove all the temporary objects that they create. It will sometimes find bugs in Venom-SC too! Look for *Garbage scanner* in the *Venom-SC Help File* index.

Finding the type of an object (new heading)

There is now a TYPEOF operator that will return the type of any Venom variable or expression – expressed as an integer in the range 0-255.

E.g.

```
TYPEOF 1.0
TYPEOF my_buffer
```

LOCKING

Deadlock page 95

Supplementary to the discussion on deadlock, note that LIST TASK and Ctrl-T (which lists tasks even when there is no command line prompt) will indicate if a task is 'blocked' waiting on a lock. It should be clear if deadlock has occurred.

Task Death page 97

The task manager now has a more efficient way of dealing with objects that might be held locked by a dead task. Please see the *Release Notes* for Venom-SC release dated 2007 01 08 and 2007 01 23 (available on our website).

ANALOGUE page 106

The Analogue object type can now drive more types of ADC device directly. Please see the *Venom-SC Help File* for full details.

Also listed there are procedures to drive yet more devices indirectly using Venom code.

BUFFER page 118

New buffer messages (new heading)

- You may implement stack-like data structures using a Buffer object and the GetLast message.
- You may remove data from the middle of a text Buffer using the Remove message.
- You can find the equivalent numerical value of the text in a buffer using the Value message.

Please see the *Venom-SC Help File* for full details of these new features.

Text Buffers page 120

The text Buffer is now not necessarily the best object to use for manipulating small amounts of text. The String object is more efficient on storage space and can perform many of the same functions. Please see the *Venom-SC Help File* for full details.

Buffer of Any (new heading)

This type of buffer can hold any object or number type, and so is useful for building lists of strings, etc. Please see the *Venom-SC Help File* for full details.

ARRAY

Non-volatile variable Arrays page 125

The syntax for these has now changed. Please see the *Venom-SC Help File* for details.

DATETIME

Spurious Dates page 132

Printing DateTime objects now can take two forms – printing 'corrected' and uncorrected forms of illegal dates such as 30th February. Please see the *Venom-SC Help File* for full details.

Nudge: this new message to DateTime allows digital watch-style alternations to individual fields within the DateTime object. Please see the *Venom-SC Help File* for full details.

ASYNCHRONOUSSERIAL page 135

A single 'software' serial port has now been implemented, adding to the original two hardware ports. This has a maximum baud rate of 9600, and has no handshaking capability. Please see the *Venom-SC Help File* for full details.

APPENDIX A: DEVELOPMENT CHECKLIST page 142

Point 8. of the checklist encourages you to test the application software. There is a tool that can help you to find errors in your program: the *Garbage scanner*.

This is an operating system tool that may be used to find 'memory leaks' – i.e. badly written programs that don't remove all the temporary objects that they create. Look for *Garbage scanner* in the *Venom-SC Help File* index.

APPENDIX B: HOW DO I ?

Store Non-Volatile Data page 143

There is now a file system based on MMC and SD memory cards. This is more secure and holds much more data than the RAM filing system. Please see the *Venom-SC Help File* for full details.

Manipulate Text page 143

The String object may be better than the text Buffer for text storage and manipulation. See Venom-SC Help File. Please see the *Venom-SC Help File* for full details.

The Protocol Analyser object may be useful for reading text from an input stream and extracting useful information such as numbers and words. Please see the *Venom-SC Help File* for full details.

Use Files page 145

There is now a file system based on MMC and SD memory cards. This is more secure and holds much more data than the RAM filing system. Please see the *Venom-SC Help File* for full details.

APPENDIX D: ROBUST APPLICATIONS page 147

Avoiding compatibility issues (new heading)

Once you have finalised your application code and tested it thoroughly, and you deem it ready to release, it may be worth 'freezing' the Application Code/Language Version combination. This avoids any possibility of problems due to changes in the Venom-SC system.

This kind of problem can arise because we are continually updating Venom-SC, which includes adding new features, modifying existing features and fixing bugs. Whilst we take care to avoid introducing bugs or incompatibilities in this process, and also regression-test each new release, it is possible that problems may occur.

The best way to freeze an Application/Language combination is to keep a master copy of your combined application/language flash chip, and only make copies from this.

Also, if you need to update your application code, then do this using the same version of the Venom-SC language.

Micro-Robotics Ltd
Cambridge CB5 8QB
+44 (0) 1223 523100
www.microrobotics.co.uk

Of course if you need a bug fix or new feature that is only available in a new version of the language then you can't adopt this practise.