Venom2 Keywords V20150903

Procedures

```
To procname … End
Return
Global
Local
AutoDestruct          Garbage collection
ParamCount            Optional parameters
Parameter(n)
```

Brackets

```
( )     Parameter list; expression grouping
[ ]     Multiple statement
```

Control flow

```
If Else
While
Do … While
Forever
Repeat
Every                 mS timed loop
Index                 Loop count...
Index0                ...from 0
Break
Select Case
```

Math Operators

```
x := y                Assignment
+ - * /
<< >>                 Shift
^                     Power
Div                   Integer division
Mod
Abs
Sqrt Exp Log
Sin Cos Tan
Asin Acos Atan
```

Logical and comparative

```
AndAlso
OrElse
IsFalse
=   <>   <   >   <=   >=
```

Bitwise operators

```
And
Or
Eor                   Exclusive Or
Inv                   Invert bits
```

Values

```
True
False
Nil                   'No object'
$1234ABCD             Hex constant
%10010111011100101    Binary constant
123.456               Float constant
123.456E10            Float constant
"Hello World"         String constant
'A'                   Character constant
\\  \"  \hh           Escape sequences
<<<:                  Embedded text –
 Literal Text         literal, mutli-line
>>>
```

Timing

```
Wait                  mS wait
Await
Every                 mS timed loop
```

Multitasking

```
Start code
Stop
Task
```

System

```
List [name, All, Task, Word]
Help
Protect(n)            Program flash
```

Data structures

```
Array
Buffer
Class
String
TextBlock
```

Comments

```
; comment             Line comment
```

Printing

```
Print
Print To
PrintF
Home
CLS
CR
BS
Beep
Chr                   Print ASCII character
Font
GotoXY                Cursor movement
Htab
Vtab
Left                  Justification
Right
Centre
~                     Print as hex
~~                    Print as binary
:                     Format specifier
```

Exception handling

```
Try
Catch
Exit
```

Data structures

```
Array
Buffer
Class
String
TextBlock
```

Objects

```
Make
New
obj.msg               Send a message
```

User-defined classes

```
Class … End
  TO method … End
This
Global
Assignment            Property setting
AutoDestruct          Garbage collection
Is                    Type checking
Has                   Interface check
Public                Accessibility
Private
Protected
Base.<msg>            Base class call
Derived.<Msg>         Late-bound call
```

Data types

```
Any
Char
Int 8                 Unsigned
Int 16                Unsigned
Int                   32 bits
Float                 IEEE single precision
String
Buffer
Class
Array
As Int, As Float, TypeOf
```

Pre-processor

```
#Define
#Define(params)
#UnDef
#Redefine
#If
#Else / #Elif
#EndIf
```

Advanced

```
@                     Form a pointer
!                     De-ref pointer
[!(pptr)(params)]     Call proc ptr
@.MsgName             Form a msg ref
obj.!(msgRef)(params) Call a msg ref
```

Built in object types

This is a list of object types built into Venom. You can also write your own types using the `Class` keyword.

```
AlphaLCD
Analogue
Array
Buffer
CANBus
CRCGenerator
DateTime
Digital
Encrypter
Ethernet
FileSystem
FTPClient
FTPServer
GraphicsLCD
HashGenerator
HTTPServer
I2Cbus
IProt
Keypad
NumberReader
OnBoardLED
OneWire
OperatingSystem
PIDController
POP3Mailbox
PPProt
PulseCounter
PulseWidthIn
PulseWidthOut
RandomNumberGen
RealTimeClock
SafeData
Semaphore
SerialPort
Shaft
SMSLink
SMTPSender
SPI
Stopwatch
String
TCProt
TextAnalyser
Timer
TouchScreen
UDProt
XMODEMLink
```

System commands

These names are interpreted as system commands. They may be used as method names, but are not recommended for other variable names.

`Debug`	*Many useful utilities*
`PrintF`	*The PrintF command*
`Protect`	*Copy the program to flash*
`Reset`	*Reset the VM2*
`Run`	*Run the program*

Operator precedence

`As Int As Float IsFalse Is Has` • *(dot)*
`! ? Abs int TypeOf Sin Cos`, *etc*
`* / ^ Div Mod >> <<`
`+ -`
`> < >= <= = <>`
`AndAlso OrElse And Or Eor`

Multi-tasking

Tasks are swapped on a Round-Robin scheme. The task swap period is nominally **1mS** per task.

Most applications use between one and six tasks. If you use more tasks than this then you may want to review your design.

In general a program is easier to write and understand if all secondary tasks are started near the start of the program, and never halt.

Signalling between tasks typically uses the task object's `State` property.

Resources used by more than one task may be locked. Most locking is handled automatically by the system, for example by the Print command.

Development

VenomIDE is the development environment for Venom. Download it for free from: www.microrobotics.co.uk

Documentation

The *Venom Tutorial* (200 pages) and *Venom2 Help File* (500 pages) are available from within VenomIDE or at www.microrobotics.co.uk

Getting started

- Set the *Prog Mode* switch to **ON** and power up the VM2 Application Board
- Type **Y** at the *Clear RAM* prompt on the VenomIDE terminal screen, to reset the VM2's program memory.

Your first program

- Open a new Venom file using VenomIDE's `File>New` menu
- Type in a new program, such as:

```
To main
  Print "Hello World", CR
End
```

- By convention the main part of a program is called **main**
- Download your program using **F7**
- Run your program using F10 (or type ***Run***).

Help

Right-clicking on any text in your program will allow you to get help on that word.

Debugging

Most debugging is done using Print statements in your code. You can also hit Ctrl-T when a program is running to show the state of each task to the terminal.

At the command line, you can run sections of your code or print out the value of any global variable.

Release

Protect your application in Flash

To protect your application from accidental erasure in the field, copy it from RAM to flash using `Protect(1)`.

Run at power-on

To have your application run at every startup, turn off the *Prog Mode* switch. If you haven't already protected your application code in Flash then you will be prompted to do so at the terminal.

Production programming

To program a batch of VM2 during production see *Production Programming* in the *Venom2 Help File*.