

COMMAND SET



[THE BLAZINGCORE SERIES]

WITH SUPERIOR NUMBER CRUNCHING ABILITIES AND PERIPHERAL HANDLING ON OUR CUSTOM EMBEDDED OS,
RAPID PROTOTYPING IS NOW EASY... AND BLAZING FAST.

BLAZINGCORE COMMAND SET

DEBUG STATEMENT

The Debug statement serves as a debugger on an embedded system. Values that are debugged using the debug statement are automatically formatted to ASCII and sent through the download port (UART2/Comm1) and displayed in the IDE's Debug Window.

`Debug.Print "[string]"` | String Variable | String Const

The Debug Statement returns Strings back to the PC. Therefore, all expressions must be in String or converted to a String (CSTR – Convert To String) before or during the instruction.

Code:

```
Dim S As String
Dim I As Integer

Public Sub Main()

    S = "This Is A Test"
    I = 5
    Debug.Print "Hello World!"
    Debug.Print S
    Debug.Print Cstr(I)

End Sub
```

Output :

```
Hello World!
This is a test
5
```

As illustrated above, all Debug Statements automatically end with a carriage return (new-line). To display String continuously on the same line, we make use of the “;” semicolon.

Code:

```
Dim S As String
Dim I As Integer

Public Sub Main()

    S = "The Number Is:"
    I = 5
    Debug.Print "Hello World!"
    Debug.Print S; " ";
    Debug.Print Cstr(I)

End Sub
```

Output :

```
Hello World!
The Number Is: 5
```

Of course, the Debug Statement accepts a mix of all the above. You may put the Cstr(I) after the semicolon on line 5.

STRING FUNCTIONS

Here is a list of functions used with the Debug.Print command to convert non-string data types to specific string formats for displaying in the debug window.

<Function> (Expression)

Function	Description
ASC	Converts the single character to its decimal representation, corresponding to its character code
Chr	Converts decimal character codes to its character representation associated with it
Cstr	Convert non-string data type variables to string
CstrBin	Convert Integer variables to its binary equivalent
CstrHex	Convert non-string data type variables to its hexadecimal equivalent in a string.

ASC

Returns the character code associated with the specified character.
Valid range for *character code* is 0 through 255.

ASC (Character)

Code:

```
DEBUG.PRINT "A = "; ASC("A")
```

Output :

```
A = 65
```

CHR

Returns the character associated with the specified character code as string.
Valid range for *character code* is 0 through 255.

Chr (Character Code)

Code:

```
DEBUG.PRINT CHR(65);" ";CHR(66);" ";CHR(67)
```

Output :

```
ABC
```

CSTR

Convert non-string data type variables to string.

Cstr(variable)

Code:

```
Dim I As Integer
Public Sub Main()
    I = 100
    Debug.Print Cstr(I)
End Sub
```

Output :

```
100
```

CSTRBIN

Convert Integer variables to its binary equivalent in a string.

`CstrBin(variable As Integer)`

Code:

```
Dim K As Integer
Public Sub Main()
    K = 54
    Debug.Print CstrBin(K)
End Sub
```

00110110

CSTRHEX

Convert non-string data type variables to its hexadecimal equivalent in a string.

`CstrHex(variable As Integer)`

Code:

```
Dim K As Integer
Public Sub Main()
    K = 15
    Debug.Print CstrHex(K)
End Sub
```

F

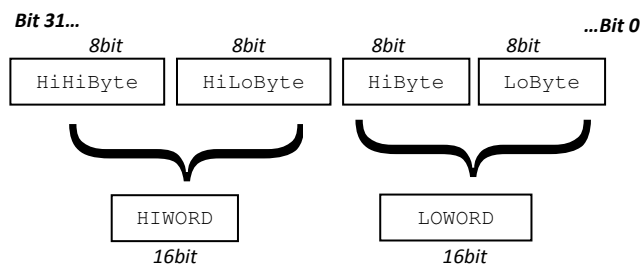
BIT, BYTE & WORD MANIPULATION

System commands are provided to enable users to manipulate values in bit, byte or word format. The following table lists the available operations and their respective description.

`<Operator>(value)`

Operator	Description
Abs	Returns the absolute value of its argument
SetBit	Set an individual bit at a specified position to logical 1. Bit Position (0 - 31) <i>Eg. SetBit(variable as integer, bit position)</i>
ClrBit	Set an individual bit at a specified position to logical 0. Bit Position (0 - 31) <i>Eg. ClrBit(variable as integer, bit position)</i>
GetBit	Get the logical state of an individual bit at a specified position.
HiByte	Returns the High Byte in the Low Word Portion of a 32bit Integer Value.
LoByte	Returns the Low Byte in the Low Word Portion of a 32bit Integer Value.
HiHiByte	Returns the High Byte in the High Word Portion of a 32bit Integer Value.
HiLoByte	Returns the Low Byte in the High Word Portion of a 32bit Integer Value.
HiWord	Returns the High Word Portion of a 32bit Integer Value.
LoWord	Returns the Low Word Portion of a 32bit Integer Value.

32bit Integer



SYSTEM DELAY

DELAY

Equivalent to `DelayMS`, where time is delayed in milliseconds.

`Delay(time)`

DELAYUS

Time is delayed in microseconds.

`DelayUS(time)`

RANDOM GENERATOR

Generate Random Integer Values.

RNDSEED()

Initialise the random generator with a seed before using it to generate values.

```
OS.RndSeed(Seed)
```

RND()

Generate random integer values with a specified range.

```
OS.Rnd(Range)
```

Code:

```
Public Sub Main()  
Dim I As Integer  
    OS.RndSeed(10)  
    I = OS.Rnd(5)  
Do  
    Debug.Print Cstr(I)  
    Delay(200)  
Loop  
End Sub
```

Output :

Output values ranging from 0-4 will be randomly generated.

LATEST DOCUMENTATION

All of our documentations are constantly updated to provide accurate and/or new information that we feel would help you with developing with our products.

The latest documentation may be obtained from our website: <http://www.aiscube.com/main/downloads.html>

HOW YOU CAN HELP

You can help us to improve our documentations by emailing to us or posting a thread in our forum, reporting any mistakes/typos or errata that you might spot while reading our documentation.

Email: TechSupport@aiscube.com

Forum: <http://forum.aiscube.com/index.php>

DISCLAIMER

All information in this documentation is provided 'as-is' without any warranty of any kind.

The products produced by AIS Cube are meant for rapid prototyping and experimental usage; they are not intended nor designed for implementation in environments that constitute high risk activities.

AIS Cube shall assume no responsibility or liability for any indirect, specific, incidental or consequential damages arising out of the use of this documentation or product.

COPYRIGHT© 2009 - 2011 AIS CUBE. ALL RIGHTS RESERVED.

ALL PRODUCT AND CORPORATE NAMES APPEARING IN THIS DOCUMENTATION MAY OR MAY NOT BE REGISTERED TRADEMARKS OR COPYRIGHTS OF THEIR RESPECTIVE COMPANIES. AND ARE ONLY USED FOR IDENTIFICATION OR EXPLANATION FOR THE OWNER'S BENEFIT. WITH NO INTENT TO INFRINGE.

SONATA IDE AND BLAZINGCORE(BCORE) ARE TRADEMARKS OF AIS CUBE IN SINGAPORE AND/OR OTHER COUNTRIES. ALL IMAGES DEPICTING THE BLAZINGCORE OR ANY PART OF IT IS COPYRIGHTED.

ALL OTHER TRADEMARKS OR REGISTERED TRADEMARKS ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS.