

AI5 CUBE

DFLASH



[THE BLAZINGCORE SERIES]










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

DFLASH

Consists of Members that enable access to the 1MB external Serial Data Flash Memory chip onboard the BlazingCore Microcontroller Board.

Note: BCore supports DFlash options of up to 16MB.

- Memory in the DFlash is organised into 16blocks of 64KB.
- Each Block consists of 16 Sectors, Each Sector is 4KB.
- Each Sector consists of 16 Pages, Each Page is 256 Bytes.

Method	Description
 Read	Read data at the specified Index in Page/Sector/Block
 Run	Runs Program Located in Specified Sector(10-15) of the DataFlash Chip
 CurrentSector	Gets Current Sector of Data Flash Chip
 CurrentPage	Gets Current Page of Data Flash Chip
 CurrentBlock	Gets Current Block of Data Flash Chip
 Sector	Provides methods to access DFlash Sectors
 Page	Provides methods to access DFlash Pages
 SizeOf	Get Size Of Specified Data Located in the DataFlash Chip
 AddressOf	Get Address Of Specified Data Located in the DataFlash Chip

ACCESSING DATA ANYWHERE IN THE DFLASH

READ

Reads a word value from the DFlash at the specified index

```
Data = DFlash.Read(ByVal Indx As Integer) As Integer
```

Method	Description
Word	Reads a Word value (16bit)
DWord	Reads a Double Word Value (32bit)

.WORD

Reads a word from the current Block, Sector and Page with the Specified Index

```
Data = DFlash.Read.Word(ByVal Indx As Integer) As Integer
```

.DWORD

Reads a double word from the current Block, Sector and Page with the Specified Index

```
Data = DFlash.Read.DWord(ByVal Indx As Integer) As Long
```

RUN

Runs Program Located in Specified Block of the DataFlash Chip

```
Result = DFlash.Run(ByVal Sector As Integer) As Integer
```

Block: Range of 10 to 15

Returns;

- -1: Operation Successful
- **All other numbers:** Error

CURRENTSECTOR

Get/Set Current Sector of Data Flash Chip

SET

```
DFlash.CurrentSector = Sector
```

GET

```
Sector = DFlash.CurrentSector()
```

Returns;

- Index of **Current Sector** As Integer

CURRENTPAGE

Get/Set Current Page of Data Flash Chip

SET

```
DFlash.CurrentPage = Page
```

GET

```
Page = DFlash.CurrentPage()
```

Returns;

- Index of **Current Page** As Integer

CURRENTBLOCK

Get/Set Current Block of Data Flash Chip

SET

```
DFlash.CurrentBlock = Block
```

GET

```
Block = DFlash.CurrentBlock()
```

Returns;

- Index of **Current Block** As Integer

SECTOR

Provides methods to access DFlash Sectors

***Note:** Use `DFlash.CurrentSector()` to get index of current sector

Method	Description
Erase	Erase Contents Of Current Sector
Read	Reads from the Current Sector at a specified Index
Write	Writes to the Current Sector at a specified Index
Copy	Reads from the Current Sector to Memory

.ERASE

Erase Contents Of Current Sector

`DFlash.Sector.Erase()`

***Note:** Use `DFlash.CurrentSector()` to get index of current sector

.READ

Reads a word from the current Block and Sector, with the Specified Index

`DFlash.Sector.Read(ByVal Index As Integer)`

Method	Description
Word	Reads a Word value (16bit)
DWord	Reads a Double Word Value (32bit)

.WORD

Reads a word from the current Block and Sector, with the Specified Index

`Data = DFlash.Sector.Read.Word(ByVal Indx As Integer) As Integer`

.DWORD

Reads a Double word from the current Block and Sector, with the Specified Index

`Data = DFlash.Sector.Read.DWord(ByVal Indx As Integer) As Long`

.WRITE

Write a Sector of data from the current Block and Sector, from a word size array ARR()

Important: A sector **must** be erased first before writing to the sector.

`DFlash.Sector.Write(ByRef ARR() As Integer)`

.COPY

Read a sector of data from the current Block and Sector, to a word size array ARR().

- Read 4096Bytes to Memory

`DFlash.Sector.Copy(ByRef ARR() As Integer/Long)`

PAGE

Provides methods to access DFlash Pages

***Note:** Use `DFlash.CurrentPage()` to get index of current Page

Method	Description
Read	Reads from the Current Page at a specified Index
Write	Writes to the Current Page at a specified Index
Copy	Reads from the Current Page to Memory

.READ

Reads a word from the current Block, Sector and Page with the Specified Index

```
DFlash.Page.Read(ByVal Index As Integer)
```

Method	Description
Word	Reads a Word value (16bit)
DWord	Reads a Double Word Value (32bit)

.WORD

Reads a word from the current Block, Sector and Page, with the Specified Index

```
Data = DFlash.Page.Read.Word(ByVal Indx As Integer) As Integer
```

.DWORD

Reads a double word from the current Block, Sector and Page with the Specified Index

```
Data = DFlash.Page.Read.DWord(ByVal Indx As Integer) As Long
```

.WRITE

Write a Page of data from the current Block and Sector, from a word size array ARR()

Important: A sector **must** be erased first before writing to the page of sector.

```
DFlash.Page.Write(ByRef ARR() As Integer)
```

.COPY

Read a Page of data from the current Block, Sector and Page, to a word size array ARR().

- Read 256Bytes to Memory

```
DFlash.Page.Copy(ByRef ARR() As Integer/Long)
```

ACCESSING DATA THROUGH THE DATA MODULES

SIZEOF

Get the size of the DATA variable from the Data Module.

Size = **DFlash.SizeOf**(DATA.VAR) As Long

Returns;

- **Size** of Specified Data

ADDRESSOF

Get the address of the DATA variable from the Data Module.

Address = **DFlash.AddressOf**(DATA.VAR) As Long

Returns;

- **Address** of Specified Data

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.