

# COP8FLASH DLL

## Using the DLL in your applications

# Table of Contents

<b>1 Introduction.....</b>	<b>3</b>
<b>1.1 Device Support.....</b>	<b>3</b>
<b>1.2 Driver Installation.....</b>	<b>3</b>
<b>1.3 Programming Language Support....</b>	<b>3</b>
<b>2 Header Support Files.....</b>	<b>4</b>
<b>2.1 Borland Delphi.....</b>	<b>4</b>
<b>2.2 Borland C++ Builder.....</b>	<b>4</b>
<b>2.3 Microsoft Visual C++.....</b>	<b>4</b>
<b>2.4 Microsoft Visual Basic.....</b>	<b>4</b>
<b>3 Exported Functions</b>	
<b>4 Programming the Device</b>	
<b>5 Support Information</b>	
<b>6 Notes</b>	

# 1

## Introduction

The COP8FLASH DLL allows the COP8 user to add ISP functionality to their own applications. This document explains how this can be achieved with a number of different programming languages.

### 1.1 Device Support

COP8CBR9	COP8CCR9	COP8CDR9
COP8SBR9	COP8SCR9	COP8SDR9
COP8AME9	COP8ANE9	COP8CBE9
COP8CCE9	COP8CDE9	COP8SBE9
COP8SCE9	COP8SDE9	COP8CFE9
COP8CBC9	COP8CCC9	COP8CDC9
COP8CFC9		

#### *DIRECTLPT version only*

COP8TAA9	COP8TAB9	COP8TAC9
----------	----------	----------

### 1.2 Driver Installation

You need to be logged on as a user with Administrator privileges to install the software under Windows NT/2000/XP.

### 1.3 Programming Language Support

The DLL should work in any Windows programming language that can call DLL files using the stdcall method, wither dynamically linked or Statically linked.

The COP8FLASH DLL has support files for Borland Delphi, Microsoft Visual C++, Borland C++ Builder and Microsoft Visual Basic.

# 2

## Header Support Files

### 2.1 Borland Delphi

To static link the dll for use with Borland Delphi you will need to add the cop8isp.pas file to your uses section. This is all that is required for the dll to be usable within Delphi.

The files to do this are available in the dlldefinitionfiles\Delphi directory.

### 2.2 Borland C++ Builder

To static link the dll for use with Borland C++ Builder you will need to include the cop8isp.h file in any cpp file that you call a dll function from.

You will also require the .lib file linked into the project. This will then allow the dll to be accessed by the project.

The files to do this are available in the dlldefinitionfiles\C++Builder directory.

### 2.3 Microsoft Visual C++

To static link the dll for use with Microsoft Visual C++ you will need to include the cop8isp.h into the headers section of the project and link the .lib file. The files to do this are available in the dlldefinitionfiles\VisualC++ directory.

### 2.4 Microsoft Visual Basic

The dll can be accessed from Visual basic using the defines declared in the file provided.

The files to do this are available in the dlldefinitionfiles\VisualBasic directory.

# 3

## Exported Functions

All functions return a non-zero value on failure.

### ***COP\_IdentifyFile(FileName : LPTSTR) : Integer;***

This function returns the type of hex file that is being used, the return value can be one of the following.

- 2 File empty or binary
- 1 File not found
- 1 (\*.hex) Intel Intellex 8/MDS
- 2 (\*.hex) Intel MCS-86
- 3 (\*.hex) Intel Hex-32
- 4 (\*.s\*) Motorola EXORciser
- 5 (\*.s\*) Motorola EXORmacs
- 6 (\*.s\*) Motorola 32-bit
- 7 Extended Tektronix
- 8 Tektronix

### ***COP\_SetDevice(DeviceName : LPTSTR) : LongWord;***

This function sets the device name that you require, a non-zero return value indicates device not found.

### ***COP\_SetMode(AMode : Byte) : LongWord;***

This function sets the programming mode that will be used for programming.

- 1 COM serial dongle programming
- 2 LPT parallel dongle programming
- 3 DirectLPT National semiconductors simple parallel interface

### ***COP\_SetPort(APort : Byte) : LongWord;***

This function sets the actual port that will be used for programming.

- 0 LPT1 COM1
- 1 LPT2 COM2
- 2 LPT3 COM3
- 3 LPT4 COM4

This function returns the number 17 if the port is not available.

# 3

## Exported Functions

***COP\_GetDevice(DeviceNum : LongWord; DeviceName : LPTSTR) : LongWord;***

This function returns the device name in the Device name variable. The DeviceNum is a number from 0 to n, n depending on the number of devices that are supported.

***COP\_GetDeviceCount : LongWord;***

This function returns the number of devices that are supported by the programmer

***COP\_EraseDevice : LongWord;***

This function bulk erases the device.

***COP\_ReadFlash(FileName : LPTSTR; FileType : Byte) : LongWord;***

This function reads the devices flash to the FileName, of file type, see the filetypes in identify file for more information.

***COP\_ProgramFlash(FileName : LPTSTR) : LongWord;***

This function programs the flash with the file provided in Filename.

***COP\_VerifyFlash(FileName : LPTSTR) : LongWord;***

This function verifies the flash with the file provided in the Filename.

***COP\_WriteOption(BT:Byte) : LongWord;***

This function writes the option byte to the device

***COP\_ReadOption : byte;***

This function returns the current option byte settings in the device

# 3

## Exported Functions

### ***COP\_WriteTiming(BT:LPTSTR) : LongWord;***

This function writes the dongle timing bytes as a string.

7.5	10mhz	settings are	7B000CFF
6	8 mhz		6F0010FF
4.5	6 mhz		630016FF
3.5	4.6mhz		5B001DFF
2.6	3.5mhz		540028FF
2	2.6mhz		4F0036FF

### ***COP\_ReadTiming : LPTSTR;***

This function returns a string containing the current set timing value in the dongle

### ***COP\_RunTarget: Longword;***

This function asserts reset and runs the device.

**NOTE** This function will only run the device if the flex bit has been set to enabled.

# 4

## Programming the Device

To program and verify a hex file the following sequence can be followed:

```
;; these functions do not connect to the device
;; and are just for the dll initialisation
COP_SetMode(1);           ;select lpt mode
COP_SetPort(0);          ; set to lpt1
COP_SetDevice('COP8CBR9'); ; device to be
                           ; programmed

; these program the chip
COP_WriteTiming('7B00CFF'); ; set to 7.5-10mhz
                           ; (see readme.txt
                           ; for values)
COP_ProgramFlash('c:\dir\flashfile.hex',);
COP_VerifyFlash('c:\dir\flashfile.hex');
COP_WriteOption(01h);     ; set flex bit
COP_RunTarget;           ; sets the flex
                           ; and disables
                           ; ISP and runs
                           ; code on chip
```

# 5

## Support Information

Technical Support for the COP8FLASH DLL is available from:

Embedded Results UK:  
support@kanda.com

If you have a bug report then please send this to the following address:  
bwhitehouse@kanda.com

Please be sure to add the software version number that you suspect you have found a bug in and the operation that you performed before you detected the bug. It may also be useful to send the cop8isp.log file that can be located in the same directory as the applications executable.

For sales queries please email:  
sales@kanda.com

# 6

## Notes

VB defines have not been tested due to not having a copy

The Visual C++ headers have not been tested

The Borland C++ headers have not been tested

***Errorcodes:***

- 0** - Operation OK
- 1** - Invalid Hardware
- 2** - Invalid Device
- 3** - Hardware Not open
- 4** - ISP Exception / Interface Connection Timeouts etc.
- 5** - Target Not Found.
- 6** - Verification Failed.
- 7** - Device is not supported on the programming system specified.
- 10** - File does not exist
- 11** - File appears corrupt
- 12** - File larger than available buffer size
- 16** - Could not create output file. (permissions, filename etc.)
- 99** - Unknown Problem..