DM6852 Driver for Windows

API Manual

For Driver Version 1.1

SWM-640010013
Rev. A

ISO9001 and AS9100 Certified

RTD Embedded Technologies, INC.
103 Innovation Blvd.
State College, PA 16803-0906

Phone: +1-814-234-8087
FAX: +1-814-234-5218

E-mail
sales@rtd.com
techsupport@rtd.com

web site
http://www.rtd.com

Revision History

08/08/2001	Version 1.0 released initial version
03/30/2004	revision A issued Documented for ISO9000

DM6852 Driver for Windows
Published by:

RTD Embedded Technologies, Inc.
103 Innovation Blvd.
State College, PA 16803-0906

Copyright 2003 RTD Embedded Technologies, Inc.
All rights reserved
Printed in U.S.A.

The RTD Logo is a registered trademark of RTD Embedded Technologies. cpuModule and utilityModule are trademarks of RTD Embedded Technologies. PC/XT, PC/AT and IBM are trademarks of International Business Machines Inc. MS-DOS, Windows, Windows 98, Windows NT, Windows 2000 and Windows XP are trademarks of Microsoft Corp. PC/104 is a registered trademark of PC/104 Consortium. All other trademarks appearing in this document are the property of their respective owners.

TABLE OF CONTENTS

LIST OF API FUNCTIONS:

Driver Initialization functions

OpenBoard6852 5
CloseBoard6852 6
InstallCallbackIRQHandler6852 7
RemoveCallbackIRQHandler6852 8

General Board Control Functions

SetBoardConfig6852 9
GetBoardConfig6852 10
ClearInterrupts6852 11
DIOReadInputData6852 12
DIOWriteStorageLatch6852 13
DIOWriteOutputLatch6852 14
DIOReadOutputLatch6852 15

PPI Control Functions

PPISetConfig6852 16
PPIGetConfig6852 17
PPIWritePort6852 18
PPIReadPort6852 19

Open

Syntax

HANDLE OpenBoard6852(ULONG DeviceNumber)

Description
This routine opens the DM6852HR board and creates handle of device file.

Parameters
[in] DeviceNumber Defines the zero based board number.

Return Value
If the function succeeds, the return value is device handle to the specified board.
If the function fails, the return value is INVALID_HANDLE_VALUE. To get extended error information, call **GetLastError**.

Example function call

```
HANDLE hBoard = OpenBoard6852(0);      // Open the first of the
                                              DM6852HR boards
…
working with the board
…
OpenBoard6852(hBoard);                 // Close the driver after
                                              work
```

CloseBoard6852

Syntax
BOOL CloseBoard6852(HANDLE hBoard)

Description
This function closes the DM6852HR board device file.

Parameters
[in] hBoard Handle to an opened DM6852HR board.

Return Value
If the function succeeds, the return value is nonzero.
If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

Example function call

```
HANDLE hBoard = OpenBoard6852(0);     // Open the first of the
                                              DM6852HR boards
…
working with the board
…
OpenBoard6852(hBoard);                // Close the driver after
                                              work
```

InstallCallbackIRQHandler6852

Syntax
BOOL InstallCallbackIRQHandler6852(HANDLE hBoard, PUSERISR pUserISR, void *pUserISRParams)

Description
This routine installs the user Interrupt Service Routine.

Parameters
[in] hBoard Handle to an open DM6852HR board.
[in] pUserISR Pointer to the user Interrupt Service Routine (ISR). This routine will be called every time, when Interrupt will occur.
[in] pUserISRParams Pointer to the user ISR parameters.

Return Value
If the function succeeds, the return value is nonzero.
If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

Example function call

```
void UserIsrIrq(void *pUserISRParams);    // user Interrupt Service
                                                Routine.
{
…
}
…
InstallCallbackIRQHandler6852(hBoard, (PUSERISR)UserIsrIrq, 0);
```

RemoveCallbackIRQHandler6852

Syntax
BOOL RemoveCallbackIRQHandler6852(HANDLE hBoard)

Description
This routine removes the user Interrupt Service Routine.

Parameters
[in] hBoard Handle to an open DM6852HR board.

Return Value
If the function succeeds, the return value is nonzero.
If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

Example function call

```
void UserIsrIrq(void *pUserISRParams);    // user Interrupt Service
                                                Routine.
{
…
}
…
InstallCallbackIRQHandler6852(hBoard, (PUSERISR)UserIsrIrq, 0);
…
RemoveCallbackIRQHandler6852(hBoard);
```

SetBoardConfig6852

Syntax
BOOL SetBoardConfig6852(HANDLE hBoard, DWORD Config)

Description
This routine sets the board configuration.

Parameters

[in] hBoard — Handle to an open DM6852HR board.

[in] Config — The control word, defining the DM6852HR board configuration.

Bit allocation table:

INTERRUPS_ENABLED_6852	Interrupts are enabled. A rising or falling edge (Depending on the trigger polarity flag) will cause an interrupt and the interrupt status bit is set.
EXTERNAL_TRIGGER_ENABLED_6852	"External trigger" is enabled. A rising or falling edge, depending on the trigger polarity flag, will clock the "Output Data Latch". This trigger will transfer the contents of the "Storage Latch" to the "Output latch ". ***Note: Software triggering is now disabled.***
EXTERNAL_TRIGGER_NON_INVERTED_68	External Trigger signal is non-inverted; rising edge will trigger.
RESET_CLEARS_DATA_6852	System reset clear Data and Output Latch disabled

Return Value
If the function succeeds, the return value is nonzero.
If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

Example function call

```
// enable IRQ and “External trigger” (“External trigger” is non-inverted)
SetBoardConfig6852(hBoard, INTERRUPS_ENABLED_6852 |
EXTERNAL_TRIGGER_NON_INVERTED_68);
```

GetBoardConfig6852

Syntax
DWORD GetBoardConfig6852(HANDLE hBoard)

Description
This routine retrieves the board configuration.

Parameters
[in] hBoard Handle to an open DM6852HR board.

Return Value
The return value is a current board configuration.
To get extended information about parameters see description of SetBoardConfig6852 routine.

Example function call
```
// check for IRQ enabled
if(GetBoardConfig6852(hBoard) & INTERRUPS_ENABLED_6852)
{
…
}
```

ClearInterrups6852

Syntax
BOOL ClearInterrupts6852(HANDLE hBoard)

Description
This routine clears the hardware interrupt line from the DM6852HR to the host.

Parameters
[in] hBoard Handle to an open DM6852HR board.

Return Value
If the function succeeds, the return value is nonzero.
If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

Example function call
ClearInterrupts6852(hBoard);

DIOReadInputData6852

Syntax
UCHAR DIOReadInputData6852(HANDLE hBoard)

Description
This function reads the optoisolation digital input from connector J8.

Parameters
[in] hBoard Handle to an open DM6852HR board.

Return Value
The return value is current state of optoisolation digital inputs. Only bits 0-3 are defined.

Example function call
UCHAR inData = DIOReadInputData6852(hBoard);

DIOWriteStorageLatch6852

Syntax
BOOL DIOWriteStorageLatch6852(HANDLE hBoard, UCHAR Data)

Description
This function writes the data to the "Storage Latch".

Parameters
[in] hBoard Handle to an open DM6852HR board.
[in] Data The data to write to the "Storage Latch".

Return Value
If the function succeeds, the return value is nonzero.
If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

Example function call
DIOWriteStorageLatch6852(hBoard, 0xFF); // write 0xFF to "Storage Latch"

DIOWriteOutputLatch6852

Syntax
BOOL DIOWriteOutputLatch6852(HANDLE hBoard)

Description
This function transfers a data from "Storage Latch" to the "Output Latch".

Parameters
[in] hBoard Handle to an open DM6852HR board.

Return Value
If the function succeeds, the return value is nonzero.
If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

Example function call
```
// transfers a data from "Storage Latch" to the "Output Latch".
DIOWriteOutputLatch6852 (hBoard);
```

DIOReadOutputLatch6852

Syntax
UCHAR DIOReadOutputLatch6852(HANDLE hBoard)

Description
Performing this function will return the status of the "Output Latch" reflecting the current state of the outputs of the MOSFETS.

Parameters
[in] hBoard Handle to an open DM6852HR board.

Return Value
The return value is the current state of the outputs of the MOSFETS.

Example function call
UCHAR outData = DIOReadOutputLatch6852(HANDLE hBoard);

PPISetConfig6852

Syntax
BOOL PPISetConfig6852(HANDLE hBoard, DWORD Config)

Description
This routine sets the PPI configuration.

Parameters
[in] hBoard Handle to an open DM6852HR board.
[in] Config Defines the PPI configuration.

Return Value
If the function succeeds, the return value is nonzero.
If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

Example function call
PPISetConfig6852(hBoard, 0x9B); // set mode 0 ports A, B and C in input mode.

PPIGetConfig6852

Syntax
DWORD PPIGetConfig6852(HANDLE hBoard)

Description
This routine retrieves the PPI configuration.

Parameters
[in] hBoard Handle to an open DM6852HR board.

Return Value
The return value is a current PPI configuration.

Example function call
DWORD ppiConfig = PPIGetConfig6852(hBoard);

PPIWritePort6852

Syntax
BOOL PPIWritePort6852(HANDLE hBoard, ULONG Port, UCHAR Data)

Description
This routine writes the PPI port.

Parameters
[in] hBoard Handle to an open DM6852HR board.
[in] Port PPI port. Available values are PORT_A, PORT_B or PORT_C.
[in] Data The data to write to specified PPI port.

Return Value
If the function succeeds, the return value is nonzero.
If the function fails, the return value is zero. To get extended error information, call **GetLastError**.

Example function call
PPIWritePort6852(hBoard, PORT_A, 0xFF);// write 0xFF to PPI port A.

PPIReadPort6852

Syntax
UCHAR PPIReadPort6852(HANDLE hBoard, ULONG Port)

Description
This routine reads the PPI port.

Parameters
[in] hBoard Handle to an open DM6852HR board.
[in] Port PPI port. Available values are PORT_A, PORT_B or PORT_C.

Return Value
Return value is a data that is read from specified port.

Example function call
UCHAR Data = PPIReadPort6852(PORT_A); // read data from PPI port A.

LIMITED WARRANTY

RTD Embedded Technologies, Inc. warrants the hardware and software products it manufactures and produces to be free from defects in materials and workmanship for one year following the date of shipment from RTD Embedded Technologies, INC. This warranty is limited to the original purchaser of product and is not transferable.

During the one year warranty period, RTD Embedded Technologies will repair or replace, at its option, any defective products or parts at no additional charge, provided that the product is returned, shipping prepaid, to RTD Embedded Technologies. All replaced parts and products become the property of RTD Embedded Technologies. Before returning any product for repair, customers are required to contact the factory for an RMA number.

THIS LIMITED WARRANTY DOES NOT EXTEND TO ANY PRODUCTS WHICH HAVE BEEN DAMAGED AS A RESULT OF ACCIDENT, MISUSE, ABUSE (such as: use of incorrect input voltages, improper or insufficient ventilation, failure to follow the operating instructions that are provided by RTD Embedded Technologies, "acts of God" or other contingencies beyond the control of RTD Embedded Technologies), OR AS A RESULT OF SERVICE OR MODIFICATION BY ANYONE OTHER THAN RTD Embedded Technologies. EXCEPT AS EXPRESSLY SET FORTH ABOVE, NO OTHER WARRANTIES ARE EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND RTD Embedded Technologies EXPRESSLY DISCLAIMS ALL WARRANTIES NOT STATED HEREIN. ALL IMPLIED WARRANTIES, INCLUDING IMPLIED WARRANTIES FOR MECHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED TO THE DURATION OF THIS WARRANTY. IN THE EVENT THE PRODUCT IS NOT FREE FROM DEFECTS AS WARRANTED ABOVE, THE PURCHASER'S SOLE REMEDY SHALL BE REPAIR OR REPLACEMENT AS PROVIDED ABOVE. UNDER NO CIRCUMSTANCES WILL RTD Embedded Technologies BE LIABLE TO THE PURCHASER OR ANY USER FOR ANY DAMAGES, INCLUDING ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES, EXPENSES, LOST PROFITS, LOST SAVINGS, OR OTHER DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT.

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES FOR CONSUMER PRODUCTS, AND SOME STATES DO NOT ALLOW LIMITATIONS ON HOW LONG AN IMPLIED WARRANTY LASTS, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

RTD Embedded Technologies, Inc.
103 Innovation Blvd.
State College PA 16803-0906
USA
Our website: www.rtd.com