Cashdrawer Function Driver

for

Beetle /Fusion Cashdrawer Interface

(Version 1.0)

TABLE OF CONTENT

1. Introduction

The Beetle /Fusion Cashdrawer Interface, which is self-explaining, support the CashDrawer control interface at the Power Supply Unit of Beetle /Fusion.

Drivers for Windows are provided in the package.

Device driver - FusionDrawer.sys is developed to access CashDrawer control interface in Windows 2000/XP.

Following files are included in the package:

File Name	*Description*
.\FusionDrawer.pdf	*This file*
.\FusionDrawerIoctl.h	*Header file for Windows application development*
.\Win\TestCard.exe	*FusionDrawer Windows Test Program*
.\Win\2K_XP\FusionDrawer.inf	*Install information file for Windows 2000 and XP*
.\Win\2K_XP\FusionDrawer.sys	*Driver for Windows 2000 and XP*

2. History of FusionDrawer Package

Version	*Date*	*Description*
1.0	7/08/2009	First released. Beta version.

3. Driver Installation

Note that Windows will not able to detect the new installed Cashdrawer Interface automatically.

3.1 Windows 2000

- Launch Add/Remove Hardware (*Start* → *Settings* → *Control Panel* → *Add/Remove Hardware*).
- Click **[Next]** to continue.
- Select "*Add/Troubleshoot a device*" and click **[Next]** .
- Select "*Add a new device*" and click **[Next]** .
- Select "*No, I want to select the hardware from a list*." and click **[Next]** .
- Select "*Other devices*" and click **[Next]** .
- Click **[Have Disk…]** . And select the path containing FusionDrawer.inf and click **[OK]** .
- Model "Beetle /Fusion Cashdrawer Interface" is prompted. Click **[Next]** to proceed.
- Click **[Next]** to install the driver.

- Click **[Finish]** to complete the installation.

3.2 Windows XP

- Launch Add Hardware (*Start* → *Settings* → *Control Panel* → *Printer and Other Hardware* → *Add Hardware*).
- Click **[Next]** to continue.
- Select "*Yes, I have already connected the hardware*." and click **[Next]** .
- Select "*Add a new hardware device*" and click **[Next]** .
- Select "*Install the hardware that I manually select from a list (Advance)*" and click **[Next]** .
- Select "*Show all devices*" and click **[Next]** .
- Click **[Have Disk…]** . And select the path containing FusionDrawer.inf and click **[OK]** .
- Model "Beetle /Fusion Cashdrawer Interface" is prompted. Click **[Next]** to proceed.
- Click **[Next]** to install the driver.
- You will then see a screen explaining that the driver you are installing has not passed the Windows Logo test. Click **[Continue Anyway]** to proceed with the installation.
- Click **[Finish]** to complete the installation.

4. Driver Testing

Simply run the test programs provided in the package.

TestCard.exe - For Windows

5. Driver Uninstallation

5.1 Windows 2000

Launch Add/Remove Hardware (*Start* → *Settings* → *Control Panel* → *Add/Remove Hardware*).

Click **[Next]** to continue.

Select "Uninstall/Unplug a device" and click **[Next]** .

Select "Uninstall a new device" and click **[Next]** .

Select "Beetle /Fusion Cashdrawer Interface" and click **[Next]** .

Select "Yes, I want to uninstall the device." and click **[Next]** .

Click **[Finish]** to complete the uninstallation.

5.2 Windows XP

Launch Device Manager (Right click *My Computer* → *Manage* → *Device Manager*).

Right click "Beetle /Fusion Cashdrawer Interface" under Other Devices, and select Uninstall

Click **[OK]** to uninstall.

6. Application Programming Interface

6.1 Windows

6.1.1 CreateFile()

Before you can access cashdrawer, you should open it from your application use standard Windows API CreateFile function:

```
HANDLE hDevice = CreateFile(  "\\\\.\\FusionDrawer0", 0, 0, 0,
                              CREATE_NEW,
                              FILE_FLAG_DELETE_ON_CLOSE,
                              0);
```

If hDevice is not equal to INVALID_HANDLE_VALUE, the driver is opened successfully.

6.1.2 DeviceIoControl()

Upon successful open drawer device, the application can issue *DeviceIoControl()* to open drawer, read drawer status, set/clear drawer status update notify.

```
CHAR   input[10];
CHAR   output[10];
DWORD nRet;
BOOL status =  DeviceIoControl( hDevice,
                                IoControlCode,
                                &input,
                                sizeof(input),
                                &output,
                                sizeof(output),
                                &nRet,
                                NULL);
```

IoControlCode supported & the possible output values are defined in header file FusionDrawerIoctl.h in the package.

Following table lists IoControlCode supported and their input/output respectively:

IoControlCode	***Function***	***Notes***
IOCTL_OPEN_DRAWER1	*Open drawer 1*	Not required for input buffer value. Output buffer length must not be less than 1. Output is set as FUSIONDRAWER_OK if successful.
IOCTL_OPEN_DRAWER2	*Open drawer 2*	

IOCTL_QUERY_DRAWER_STATUS	*Read drawer status*	Not required for input buffer value. Output buffer length must not be less than 1.
		After issuing IOCTL_QUERY_DRAWER_ STATUS, use following code to parse the output:
		if (output[0] & FUSIONDRAWER_1)
		// Drawer 1 is close
		else
		// Drawer 1 is open
		
		if (output[0] & FUSIONDRAWER_2)
		// Drawer 2 is close
		else
		// Drawer 2 is open
		Note: FUSIONDRAWER_1, FUSIONDRAWER_2 are defined in FusionDrawerIoctl.h.

IOCTL_SET_NOTIFY	*Set status update notify to allow driver to fire event to notify appli-cation when drawer status is changed.*	Input buffer contains handle to event to be notified. Output buffer length must not be less than 1. Please refer to following sample code:
		PVOID inBuf[1];
		CHAR outBuf[1];
		HANDLE hSyncEvent = CreateEvent(NULL, FALSE, FALSE, NULL);
		inBuf[0] = hSyncEvent;
		if (! DeviceIoControl(hDevice, IOCTL_SET_NOTIFY, inBuf, sizeof(PVOID), outBuf, sizeof(outBuf), &nRet, NULL))
		// DeviceIoControl fail.

IOCTL_CLEAR_NOTIFY	*Clear status update notify set earlier.* *Disable driver event firing to notify appli-cation when drawer status is changed.*	Not required for input buffer value. Output buffer length must not be less than 1. Output is set as FUSIONDRAWER_OK if successful.

6.1.3 CloseHandle()

At the end of the application program, the board must be closed with the Windows standard API CloseHandle() function:

CloseHandle(hDevice);

7. Appendix – FusionDrawerIoctl.h

```
//==========================================================================/
// FILE:              $Workfile: FusionDrawerIoctl.h $
// AUTHOR(OWNER):     Jason Chew, created 09.07.09
// ---------------------------------------------------------------------------
// Copyright © by Wincor Nixdorf Pte Ltd, Singapore in 2009
// ---------------------------------------------------------------------------
// SHORT DESCRIPTION:
//
// FusionDrawerIoctl.h - DeviceIoControl IOCTL codes supported by Beetle /Fusion
//                       Cashdrawer interface driver.
//
// VERSION: 1.0.<revision version>
//
//==========================================================================/
//==================== END OF HEADER  ==============================//
//
// HISTORY (File history):
//
// $History: FusionDrawerIoctl.h $
// *****************  Version 2  *****************
// User: Jason Chew   Date: 06.08.09  Time: 16:30
// Updated to support 2 cashdrawers connected to same Cashdrawer Interface.
//
// *****************  Version 1  *****************
// User: Jason Chew   Date: 09.07.09  Time: 13:30
// Create new version for BEETLE /FUSION SMBus Cashdrawer Interface.
//
////////////////////////////////////////////////////////////////////////////
// Output return codes

enum FUSION_DRAWER_IO_ERRORS
{
        FUSION_DRAWER_OK = 0,                       // Complete successfully

        FUSION_DRAWER_CANNOT_CONNECT_TO_INTERRUPT,
         // IRQ_CONNECT: cannot connect to the given interrupt

        FUSION_DRAWER_WRONG_BUFFER_LEN,     // Buffer too small
        FUSION_DRAWER_EVENT_STILL_ACTIVE,   // Status update notify is set
already
        FUSION_DRAWER_IOCTL_ABORT,          // IoControl aborted
        FUSION_DRAWER_PASS_EVENT_FAIL       // Fail to pass event to be notified
};

////////////////////////////////////////////////////////////////////////////
//
// Status Mask

#define FUSION_DRAWER_1     0x01
#define FUSION_DRAWER_2     0x01

// Open drawer 1
#define IOCTL_OPEN_DRAWER1 CTL_CODE(                \
                           FILE_DEVICE_UNKNOWN,     \
```

```
                    0x801,                          \
                    METHOD_BUFFERED,                \
                    FILE_ANY_ACCESS)

// Open drawer 2
#define IOCTL_OPEN_DRAWER2 CTL_CODE(                \
                    FILE_DEVICE_UNKNOWN,            \
                    0x802,                          \
                    METHOD_BUFFERED,                \
                    FILE_ANY_ACCESS)

// Read status of both drawers
#define IOCTL_QUERY_DRAWER_STATUS CTL_CODE(         \
                    FILE_DEVICE_UNKNOWN,            \
                    0x803,                          \
                    METHOD_BUFFERED,                \
                    FILE_ANY_ACCESS)

// Set drawer status update notify
#define IOCTL_SET_NOTIFY CTL_CODE(                  \
                    FILE_DEVICE_UNKNOWN,            \
                    0x804,                          \
                    METHOD_BUFFERED,                \
                    FILE_ANY_ACCESS)

// Clear drawer status update notify
#define IOCTL_CLEAR_NOTIFY CTL_CODE(                \
                    FILE_DEVICE_UNKNOWN,            \
                    0x805,                          \
                    METHOD_BUFFERED,                \
                    FILE_ANY_ACCESS)

// Query Version
#define IOCTL_QUERY_VERSION CTL_CODE(               \
                    FILE_DEVICE_UNKNOWN,            \
                    0x806,                          \
                    METHOD_BUFFERED,                \
                    FILE_ANY_ACCESS)
```