ECDR212/214 NT Driver

Software Design Document

ECHOTEK CORPORATION
555 Sparkman Drive, Suite 400
Huntsville, AL 35816
Phone: (205) 721-1911
Fax: (205) 721-9266
Web Site: http://www.echotek.com

TABLE OF CONTENTS

1) INTRODUCTION **3**

2) USER-LEVEL FUNCTIONS **4**

EC_ECDR212_CLOSE 4
PURPOSE *4*
SYNOPSIS *4*
DESCRIPTION *4*
RETURN VALUES *4*
EC_212_IOCTL 5
PURPOSE *5*
SYNOPSIS *5*
DESCRIPTION *5*
RETURN VALUES *6*
EC_212_OPEN 7
PURPOSE *7*
SYNOPSIS *7*
DESCRIPTION *7*
RETURN VALUES *7*
EC_ECDR212_READ 8
PURPOSE *8*
SYNOPSIS *8*
DESCRIPTION *8*
RETURN VALUES *8*

3) GLOBAL VARIABLE REFERENCE **9**

4) STRUCTURE REFERENCE **10**

EC_ECDR212_REG_HANDLE 10
EC_HANDLE 10
EC_ECDR212_INFO_T 11
EC_ECDR212_INFO_T 11

5) FILE REFERENCE **12**

6) ECDR212 DRIVER SUPPORT **13**

A) REFERENCES **14**

1) Introduction

Echotek's ECDR212/214 NT driver, henceforth referred to as the ECDR212 driver, will facilitate ease of use of Echotek's ECDR212 and/or ECDR214. This driver will be supplied as a statically linked library that will provide the following APIs:

a) *EC_212_Close* – closes the ECDR212
b) *EC_212_Ioctl* – performs various informative operations
c) *EC_212_Open* – opens the ECDR212 for reading
d) *EC_212_Read* – performs a synchronous read from the ECDR212

For a more detailed description, see section 2) User-Level Functions.
For hardware-level information see the hardware manual [ECH99] or [ECH00].

2) User-Level Functions

The functions that will allow the user to utilize the ECDR212 will be EC_212_Open, EC_212_Close, EC_212_Read, and EC_212_Ioctl. These functions, in conjunction with ec_shared.lib and ec_util.lib, will facilitate the use of the ECDR212 and will make up the core of the ECDR212 driver.

EC_ECDR212_Close

PURPOSE

```
To release the resources associated with a particular ECDR212.
```

SYNOPSIS

```
#include <EC_ECDR212.h>

EC_HANDLE EC_212_Close(EC_HANDLE hEC);
```

DESCRIPTION

```
EC_212_Close releases the resources associated with a particular
ECDR212.

Exits immediately if hEC is NULL.
```

RETURN VALUES

```
NULL
```

EC_212_Ioctl

PURPOSE

To perform miscellaneous access to and control of a particular ECDR212. EC_212_Ioctl takes a variable number of arguments as described below.

SYNOPSIS

```
#include <EC_ECDR212.h>

STATUS EC_212_Ioctl(EC_HANDLE    hEC,
                    DWORD        Command,
                    DWORD        Arg1,
                    DWORD        Arg2,
                    ...);
```

DESCRIPTION

EC_212_Ioctl performs miscellaneous access to and control of a particular ECDR212. Primarily used to support debug efforts; however, EC_212_Ioctl must be used as described below to program the board.

The following commands are available:

EC_ECDR212_PROGRAM – programs the ECDR212; Arg1 is an initialization file name (see section **Error! Reference source not found.-Error! Reference source not found.**); no other arguments are used

EC_CFG_READ - reads a PCI configuration register; Arg1 is the register offset to be read; Arg2 is an address to which the register contents will be stored; no other arguments are used

EC_PROBE_PLX - reads a PLX register; Arg1 may be READ which indicates that a register will be read or WRITE which indicates that a register will be written; Arg2 is the register offset to be either read or written; Arg3 is either an address to which the register contents will be stored or the address of the value to be written to the register; no other arguments are used

EC_PROBE_ECDR212 - reads an ECDR212 register; Arg1 may be READ which indicates that a register will be read or WRITE which indicates that a register will be written; Arg2 is the register offset to be either read or written; Arg3 is either

```
                              an address to which the register
                              contents will be stored or the
                              address of the value to be written
                              to the register; no other arguments
                              are used
          EC_PROBE_CH1_FIFO - reads one word from the channel
                               1 FIFO; Arg1 is an address to
                               which the FIFO contents will be
                               stored; no other arguments are
                               used
          EC_PROBE_CH2_FIFO - reads one word from the channel
                               2 FIFO; Arg1 is an address to
                               which the FIFO contents will be
                               stored; no other arguments are
                               used
```

RETURN VALUES

On success, EC_OKAY.

Otherwise, EC_ERROR is returned.

EC_212_Open

PURPOSE

To allocate the resources associated with a particular ECDR212.

SYNOPSIS

```
#include <EC_ECDR212.h>

EC_HANDLE EC_212_Open(DWORD Instance);
```

DESCRIPTION

EC_212_Open allocates the resources associated with a particular instance of an ECDR212. This Instance is simply an integer. For example:

```
A system with 1 ECDR212: Instance is always 0

A system with 2 ECDR212s: Instance is 0 for the ECDR212 in
                          the first PMC site and 1 for the
                          ECDR212 in the second PMC site.

A system with n ECDR212s: Instance is 0 for the ECDR212 in
                          the first PMC site, 1 for the
                          ECDR212 in the second PMC site,
                          and n-1 for the ECDR212 in the
                          nth PMC site.
```

RETURN VALUES

On success, an EC_HANDLE to a particular instance of an ECDR212.

Otherwise, NULL is returned.

EC_212_Read

PURPOSE

To read data synchronously from an ECDR212.

SYNOPSIS

```
#include <EC_ECDR212.h>

DWORD EC_212_Read(EC_HANDLE hEC,
                  DWORD     *Buffer_Address,
                  DWORD     Word_Count,
                  DWORD     Channel_No);
```

DESCRIPTION

EC_212_Read reads Word_Count from Channel_No associated with hEC to Buffer_Address. The data format is dependent on the data order switch on the board. When position 1 of this switch is “ON”, the data is read from the board with the I as the lower word and the Q as the upper word. This data order is reversed if position 1 is “OFF”. See the hardware manual [ECH99] or [ECH00] for more details.

RETURN VALUES

On success, EC_OKAY is returned.

Otherwise, EC_ERROR is returned.

3) Global Variable Reference

The following global variables will be declared in the file EC_212_Globals.cpp.

```
EC_PROG_INFO_t      Default_Prog_Info; /* Default ECDR212 programming
                                          information */
EC_PROG_INFO_t      Close_Prog_Info;   /* Programming information for
                                          closing the ECDR212 */
DWORD               FIFO_STAT;         /* The status of the FIFO */
EC_HANDLE           gEC;               /* A handle to the driver */
```

4) Structure Reference

EC_ECDR212_REG_HANDLE

EC_ECDR212_REG_HANDLE will map the register set of the ECDR212 to memory. For a more detailed description of the ECDR212 registers, see the ECDR212 Reference Manual [ECH99].

```
typedef struct EC_ECDR212_REG_STRUCT {
  long Command_Status;                    /* 00 */
  long Rcvr_Mode;                         /* 04 */
  long Trig_Command;                      /* 08 */
  long Rcvr_Decimation;                   /* 0C */
  long Rcvr_Burst_Count;                  /* 10 */
  long unused[11];                        /* 14 - 3C */
  long Int_Control;                       /* 40 */
  long Int_Status;                        /* 44 */
  long FIFO_Status;                       /* 48 */
  long Clk_Control;                       /* 4C */
  long Reset_Cmd;                         /* 50 */
  long Timer_Reg;                         /* 54 */
  long Test_Reg;                          /* 58 */
  long unused2[16361];                    /* 0005C - 0FFFC */
  RCVR_6620_t RCVR_CH1;                   /* 10000 - 1FFFC */
  RCVR_6620_t RCVR_CH2;                   /* 20000 - 2FFFC */
}EC_ECDR212_REG_t, *EC_ECDR212_REG_HANDLE;
```

EC_HANDLE

EC_HANDLE is a driver-related pointer to a structure containing ECDR212 information. It is used to keep track of information about current ECDR212 usage. *EC_HANDLE* will be used whenever a driver function is called. *EC_ECDR212_Open* allocates and initializes *EC_HANDLE* and *EC_ECDR212_Close* frees it.

```
typedef struct EC_STRUCT
{
        EC_PCI_Handle                   hEC_PCI;
        PLX_9080_Handle                 hP9080;
        EC_ECDR212_REG_HANDLE           h       ECDR212_Reg;
        EC_PROG_INFO_t                   Prog_Info;
        DWORD                           Trig_Mode;
        DWORD                           Is_Busy;
        DWORD                           Is_Quad;
        EC_PCI_DESC                     EC_PCI_Desc;
        WD_TRANSFER                     Transfer[7];
}EC_t, *EC_HANDLE;
```

EC_ECDR212_INFO_t

EC_ECDR212_INFO_t is structure containing ECDR212 register programming information

```
/*---[ ECDR212 Register Programming Information ]---*/
typedef struct
{
  long Command_Status;
  long Rcvr_Mode;
  long Rcvr_Decimation;
  long Rcvr_Burst_Count;
  long Int_Control;
  long Clk_Control;
  long Timer_Reg;
} EC_ECDR212_INFO_t;
```

EC_ECDR212_INFO_t

EC_PROG_INFO_HANDLE is a structure containing ECDR212 programming information

```
/*---[ ECDR212 Programming Information ]---*/
typedef struct
{
  long                    Is_Valid;
  EC_ECDR212_INFO_t       ECDR212_Info;
  RCVR_6620_INFO_t        RCVR_Info[2];
}EC_PROG_INFO_t, *EC_PROG_INFO_HANDLE;
```

5) File Reference

The following files will contain the source code for the ECDR212 driver:

EC_ECDR212.h
EC_ECDR212_Error.h
EC_ECDR212_Prototypes.h
EC_212_Close.c
EC_212_Globals.c
EC_212_Int_Enable.c
EC_212_Ioctl.c
EC_212_ISR.c
EC_212_Open.c
EC_212_Program.c
EC_212_Read.c
EC_6620_Program.c
EC_9080_Init.c
EC_FIFO_Probe.c
EC_Get_6620_Info_New.c
EC_Get_Option.c
EC_Prog_Init.c
EC_Scan_File.c

In addition, ec_shared.lib and ec_util.lib will be used.

6) ECDR212 Driver Support

Support can be obtained by contacting the software department via one of the following methods:

e-mail:	jane@echotek.com	
Phone:	(256)721-1911	Attn: Jane Childers
Fax:	(256)721-9266	Attn: Jane Childers
Address:	Echotek Corporation Attn: Software Department 555 Sparkman Drive, Suite 400 Huntsville, AL 35816	

A) References

[ECH99] Echotek Corporation, "ECDR212 Reference Manual", 1999.
[ECH00] Echotek Corporation, "ECDR214 Reference Manual", 2000.