
ReadTheDocs-Breathe Documentation

Release 1.0.0

Thomas Edvalson

Aug 22, 2017

Contents

1 Indices and tables

3

Contents:

- [genindex](#)
- [modindex](#)
- [search](#)

```
struct usb_cdc_line_coding  
    #include <cdc.h>
```

Public Members

```
    __u8 bDTERate0  
    __u8 bDTERate1  
    __u8 bDTERate2  
    __u8 bDTERate3  
    __u8 bCharFormat  
    __u8 bParityType  
    __u8 bDataBits
```

file **lights.c**

```
#include <fx2regs.h>#include <fx2types.h>#include <delay.h>#include "fx2lights.h" Example firmware that  
flashes LEDs.
```

On the Atlys this is LD14 (DONE), on the FX2 miniboard D1 and D2 and on the Ophis it is D2

Functions

```
void init_lights (void)  
void main (void)
```

file **serial.c**

```
#include <delay.h>#include <fx2macros.h>#include <fx2regs.h>#include <fx2types.h>
```

Defines

PD3

BAUD

Functions

void **usart_init** (void)

void **usart_send_byte** (BYTE *c*)

void **usart_send_string** (const char **s*)

void **main** (void)

Variables

__sbit __at PD3 USART

file **app.c**

```
#include <fx2regs.h>#include <fx2macros.h>#include <eutils.h>#include <delay.h>#include <setup-dat.h>#include <makestuff.h>#include "defs.h"#include "debug.h"#include "cdc.h"#include "uvc.h"#include "read-serialno.h"
```

Functions

void **TD_Init** (void)

void **TD_Poll** (void)

void **uart_init** ()

void **mainInit** (void)

void **mainLoop** (void)

uint8 **handleVendorCommand** (*uint8 cmd*)

Variables

const *uint8* **dev_strings** []

file **cdc-config.h**

Defines

CDC_H2D_RESET (x)

CDC_H2D_EP (x)


```

bmCDC_H2D_EP (x)
CDC_D2H_RESET (x)
CDC_D2H_EP (x)
bmCDC_D2H_EP (x)
CDC_INT_EP (x)
bmCDC_INT_EP (x)

```

file **cdc.c**

```
#include <fx2regs.h>#include <fx2macros.h>#include <delay.h>#include "cdc.h"
```

Defines

```
SYNCDELAY
```

Functions

```

void cdc_receive_poll ()
BOOL cdc_handle_command (BYTE cmd)

```

Variables

```

volatile WORD cdc_queued_bytes = 0
struct usb_cdc_line_coding cdc_current_line_coding = { .bDTERate0 = LSB(2400), .bDTERate1 = MSB(2400), .bDT

```

file **cdc.h**

```
#include <fx2types.h>#include "cdc-config.h"
```

Defines

```

cdc_can_send
cdc_queue_data (data)
cdc_send_queued_data
__u8
    The defines and structures found below comes from the Linux kernel and are found in include/uapi/linux/usb/cdc.h
__le16
__le32
USB_CDC_REQ_SET_LINE_CODING
USB_CDC_REQ_GET_LINE_CODING
USB_CDC_REQ_SET_CONTROL_LINE_STATE
USB_CDC_1_STOP_BITS
USB_CDC_1_5_STOP_BITS

```

USB_CDC_2_STOP_BITS
USB_CDC_NO_PARITY
USB_CDC_ODD_PARITY
USB_CDC_EVEN_PARITY
USB_CDC_MARK_PARITY
USB_CDC_SPACE_PARITY

Functions

void `cdcuser_receive_data` (BYTE **data*, WORD *length*)
BOOL `cdcuser_set_line_rate` (DWORD *baud_rate*)
BOOL `cdc_handle_command` (BYTE *cmd*)
void `cdc_receive_poll` ()

Variables

volatile WORD `cdc_queued_bytes`

file `debug.c`
 #include <fx2regs.h>#include <makestuff.h>#include "debug.h"

file `debug.h`
 #include <makestuff.h>

Functions

void `usartInit` (void)
void `usartSendByte` (*uint8 c*)
void `usartSendByteHex` (*uint8 byte*)
void `usartSendWordHex` (*uint16 word*)
void `usartSendLongHex` (*uint32 word*)
void `usartSendString` (const char **s*)

file `defs.h`

Defines

SYNCDELAY
EPOBUF_SIZE
bmSKIP
bmPORTS
bmGPIF

bmFIFOS
bmBULK
bmBUF2X
bmEP1INBSY
bmEP1OUTBSY
bmAPTREN
bmAPTR1INC
bmAPTR2INC
bmDYN_OUT
bmENH_PKT
REQDIR_DEVICETOHOST
REQDIR_HOSTTODEVICE
REQTYPE_CLASS
REQTYPE_STANDARD
REQTYPE_VENDOR
mainInit (x)
mainLoop (x)
handleVendorCommand

Variables

__sfr __at AUTODAT1
__sfr __at AUTODAT2

file **firmware.c**

```
#include <fx2macros.h>#include <fx2ints.h>#include <autovector.h>#include <delay.h>#include <setup-dat.h>#include "cdc.h"
```

Defines

printf (...)

Copyright (C) 2009 Ubixum, Inc.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

SYNCDELAY

Functions

```
void main_loop ()  
void main_init ()  
void main ()  
void resume_isr ()  
void sudav_isr ()  
void usbreset_isr ()  
void hispeed_isr ()  
void suspend_isr ()  
void ISR_USART0 (void)
```

Variables

```
volatile __bit dosud =FALSE  
volatile __bit dosuspend =FALSE
```

```
file makestuff.h  
#include <stddef.h>#include <stdbool.h>
```

Defines

```
WARN_UNUSED_RESULT  
DLLEXPORT (t)  
PFSZD  
PFSZH  
WORD_LENGTH  
NULL  
swap16 (x)  
CONCAT_INTERNAL (x, y)  
CONCAT (x, y)  
STR_INTERNAL (x)  
STR (x)  
VA_EXPAND (x)  
VA_NARGS (...)  
VA_NARGS_INTERNAL (a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, N, ...)  
FAIL (code, label)  
CHECK_INTERNAL3 (condition, code, label)  
CHECK_INTERNAL4 (condition, code, label, prefix)
```

CHECK_INTERNAL5 (condition, code, label, ...)

CHECK_STATUS (...)

Typedefs

typedef unsigned char **uint8**

typedef unsigned short **uint16**

typedef unsigned long long **uint64**

typedef signed char **int8**

typedef signed short **int16**

typedef signed int **int32**

typedef unsigned int **uint32**

typedef signed long long **int64**

typedef unsigned int **bitfield**

file **read-serialno.c**
#include <i2c.h>

Defines

PROM_ADDRESS

PROM_ID_OFFSET

PROM_ID_SIZE

Functions

char **hex** (BYTE *value*)

void **patch_serial_number** (BYTE *index*, BYTE *value*)

void **patch_usb_serial_number_with_eeprom_macaddress** ()

Variables

__xdata char **dev_serial** []

file **read-serialno.h**

Functions

void **patch_usb_serial_number_with_eeprom_macaddress** ()

file **softserial.c**
#include "debug.h" *#include* "softserial.h"

Functions

void **soft_sio0_init** (WORD *baud_rate*)

void **soft_putchar** (char *c*)

char **soft_getchar** (void)

file **softserial.h**
#include "fx2types.h"

Functions

void **soft_sio0_init** (WORD *baud_rate*)

void **soft_putchar** (char *c*)

char **soft_getchar** ()

file **to-uart.c**
#include <stdio.h>*#include* <fx2regs.h>*#include* <fx2macros.h>*#include* <serial.h>*#include* <delay.h>*#include* <autovector.h>*#include* <setupdat.h>*#include* <eputils.h>*#include* <i2c.h>*#include* "cdc.h"

Defines

SYNCDELAY

Copyright (C) 2009 Ubixum, Inc.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Functions

BOOL **cdcuser_set_line_rate** (DWORD *baud_rate*)

void **cdcuser_receive_data** (BYTE **data*, WORD *length*)

void **uart_init** ()

file **uvc.c**
#include "uvc.h"*#include* <fx2regs.h>*#include* <fx2macros.h>*#include* <setupdat.h>*#include* <eputils.h>*#include* <delay.h>*#include* "cdc-config.h"

Defines

SYNCDELAY

Functions

```

BOOL handleUVCCommand (BYTE cmd)
BYTE handle_get_configuration ()
BOOL handle_set_configuration (BYTE cfg)
BOOL handle_get_interface (BYTE ifc, BYTE *alt_ifc)
BOOL handle_set_interface (BYTE ifc, BYTE alt_ifc)
BOOL handle_get_descriptor ()

```

Variables

```

    0x00,0x00, 0x00,0x00, 0x00,0x00, 0x00,0x00, 0x05,0x00, 0x00,0x20,0x1C,0x00, 0x00,0x04,0x00,0x00
    }]
BYTE fps[2][4] = {{0x2A,0x2C,0x0A,0x00},{0x54,0x58,0x14,0x00}}
BYTE frameSize[2][4] = {{0x00,0x00,0x18,0x00},{0x00,0x20,0x1C,0x00}}
BYTE Configuration
BYTE AlternateSetting = 0

```

```

file uvc.h
#include <fx2types.h>

```

Defines

```

UVC_GET_CUR
UVC_GET_MIN
UVC_GET_MAX

```

Functions

```

BOOL handleUVCCommand (BYTE cmd)

```

```

file README.md
dir examples
dir hdmi2usb
dir examples/lights
dir examples/serial
page index

```

This repository contains the firmware used in the [HDMI2USB project](#) on the Cypress FX2 chip found on boards such as the [Numato Opsis](#) and the [Digilent Atlys](#) boards.

Firmware is generally compiled with [SDDC 3.6]() and the sdas8051 assembler. Older versions *might* work but have been known to generate incorrect firmware.

microload is firmware which loads a second stage firmware from an FPGA via I2C.

In microload every program **byte** counts and thus is hand coded in assembly.

The firmware is used on the Numato Opsi board to allow the FPGA to switch the FX2 functionality.

eprom-unconfigured will cause the Cypress FX2 which boot with a given VID/PID and then wait for full firmware to be loaded via USB.

The JTAG firmware is used to allow loading gateway onto an FPGA.

- [ixo-usb-jtag](#)

This is the primary firmware for HDMI2USB functionality.

The firmware is responsible for transporting the video data off of the FPGA. It does this by enumerating as a USB Video Class device; a standard way of interfacing devices such as webcams to a host. Linux, Windows and OS X all include support out of the box for reading taking to such devices.

The firmware is also used for control and debugging of the system, through a USB Communications Class Device. This is a common way of attaching serial ports to the system, and under Linux it can be accessed at `/dev/ttyUSBX`.

Cypress EZ-USB FX2LP

FIXME: Put some documentation about the FX2 here.

[Cypress EZ-USB FX2LP](#)

Boards

- FX2 - CY7C68013A_100AC - 100 pin version
- EEPROM - 24AA02E48 - 256*bytes* (128 usable)
- Bank A/Port A, Bank B/Port B, Bank C/Port D, Bank F/Port C, Bank H/RDY+CTRL - FPGA IO Pins
- Bank G/Port E connected to FPGA JTAG.
- 2 x Hardware UARTs
- 1 x Interrupts?
- [Further Documentation](#)

- FX2 - CY7C68013A-56 - 56 pin version
- EEPROM - 24AA128 - 16k*bytes*
- Bank A/Port A, Bank B/Port B, RDY+CTRL - FPGA IO Pins
- Bank C/Port D connected to FPGA JTAG.
- [Further Documentation](#)

Building


```
make -f Makefile.fx2 load-fx2
```

will build and flash the FX2 firmware. This requires the `hdmi2usb-mode-switch` command, which on Debian Sid and Stretch can be installed using the `hdmi2usb-mode-switch` package, otherwise see [HDMI2USB-mode-switch](#).

There are three ways to run `hdmi2usb-mode-switch`:

- 1.As root
- 1.Install the `unbind-helper` as a setuid binary
- 1.Install the `udev-rules` which sets the permissions of the unbind.

Symbols

`__le16` (C macro), 5
`__le32` (C macro), 5
`__u8` (C macro), 5

A

AlternateSetting (C++ member), 11

B

BAUD (C macro), 4
bitfield (C++ type), 9
bmAPTR1INC (C macro), 7
bmAPTR2INC (C macro), 7
bmAPTREN (C macro), 7
bmBUF2X (C macro), 7
bmBULK (C macro), 7
bmCDC_D2H_EP (C macro), 5
bmCDC_H2D_EP (C macro), 4
bmCDC_INT_EP (C macro), 5
bmDYN_OUT (C macro), 7
bmENH_PKT (C macro), 7
bmEP1INBSY (C macro), 7
bmEP1OUTBSY (C macro), 7
bmFIFOS (C macro), 6
bmGPIF (C macro), 6
bmPORTS (C macro), 6
bmSKIP (C macro), 6

C

cdc_can_send (C macro), 5
cdc_current_line_coding (C++ member), 5
CDC_D2H_EP (C macro), 5
CDC_D2H_RESET (C macro), 5
CDC_H2D_EP (C macro), 4
CDC_H2D_RESET (C macro), 4
cdc_handle_command (C++ function), 5, 6
CDC_INT_EP (C macro), 5
cdc_queue_data (C macro), 5
cdc_queued_bytes (C++ member), 5, 6

cdc_receive_poll (C++ function), 5, 6
cdc_send_queued_data (C macro), 5
cdcuser_receive_data (C++ function), 6, 10
cdcuser_set_line_rate (C++ function), 6, 10
CHECK_INTERNAL3 (C macro), 8
CHECK_INTERNAL4 (C macro), 8
CHECK_INTERNAL5 (C macro), 8
CHECK_STATUS (C macro), 9
CONCAT (C macro), 8
CONCAT_INTERNAL (C macro), 8
Configuration (C++ member), 11

D

dev_strings (C++ member), 4
DLLEXPORT (C macro), 8
dosud (C++ member), 8
dosuspend (C++ member), 8

E

EPOBUF_SIZE (C macro), 6

F

FAIL (C macro), 8
fps (C++ member), 11
frameSize (C++ member), 11

H

handle_get_configuration (C++ function), 11
handle_get_descriptor (C++ function), 11
handle_get_interface (C++ function), 11
handle_set_configuration (C++ function), 11
handle_set_interface (C++ function), 11
handleUVCCCommand (C++ function), 11
handleVendorCommand (C macro), 7
handleVendorCommand (C++ function), 4
hex (C++ function), 9
hispeed_isr (C++ function), 8

I

init_lights (C++ function), 3

int16 (C++ type), 9
int32 (C++ type), 9
int64 (C++ type), 9
int8 (C++ type), 9
ISR_USART0 (C++ function), 8

M

main (C++ function), 3, 4, 8
main_init (C++ function), 8
main_loop (C++ function), 8
mainInit (C macro), 7
mainInit (C++ function), 4
mainLoop (C macro), 7
mainLoop (C++ function), 4

N

NULL (C macro), 8

P

patch_serial_number (C++ function), 9
patch_usb_serial_number_with_eeprom_macaddress
(C++ function), 9
PD3 (C macro), 4
PFSZD (C macro), 8
PFSZH (C macro), 8
printf (C macro), 7
PROM_ADDRESS (C macro), 9
PROM_ID_OFFSET (C macro), 9
PROM_ID_SIZE (C macro), 9

R

REQDIR_DEVICETOHOST (C macro), 7
REQDIR_HOSTTODEVICE (C macro), 7
REQTYPE_CLASS (C macro), 7
REQTYPE_STANDARD (C macro), 7
REQTYPE_VENDOR (C macro), 7
resume_isr (C++ function), 8

S

soft_getchar (C++ function), 10
soft_putchar (C++ function), 10
soft_sio0_init (C++ function), 10
STR (C macro), 8
STR_INTERNAL (C macro), 8
sudav_isr (C++ function), 8
suspend_isr (C++ function), 8
swap16 (C macro), 8
SYNCDELAY (C macro), 5–7, 10

T

TD_Init (C++ function), 4
TD_Poll (C++ function), 4

U

uart_init (C++ function), 4, 10
uint16 (C++ type), 9
uint32 (C++ type), 9
uint64 (C++ type), 9
uint8 (C++ type), 9
usart_init (C++ function), 4
usart_send_byte (C++ function), 4
usart_send_string (C++ function), 4
usartInit (C++ function), 6
usartSendByte (C++ function), 6
usartSendByteHex (C++ function), 6
usartSendLongHex (C++ function), 6
usartSendString (C++ function), 6
usartSendWordHex (C++ function), 6
USB_CDC_1_5_STOP_BITS (C macro), 5
USB_CDC_1_STOP_BITS (C macro), 5
USB_CDC_2_STOP_BITS (C macro), 5
USB_CDC_EVEN_PARITY (C macro), 6
usb_cdc_line_coding (C++ class), 3
usb_cdc_line_coding::bCharFormat (C++ member), 3
usb_cdc_line_coding::bDataBits (C++ member), 3
usb_cdc_line_coding::bDTERate0 (C++ member), 3
usb_cdc_line_coding::bDTERate1 (C++ member), 3
usb_cdc_line_coding::bDTERate2 (C++ member), 3
usb_cdc_line_coding::bDTERate3 (C++ member), 3
usb_cdc_line_coding::bParityType (C++ member), 3
USB_CDC_MARK_PARITY (C macro), 6
USB_CDC_NO_PARITY (C macro), 6
USB_CDC_ODD_PARITY (C macro), 6
USB_CDC_REQ_GET_LINE_CODING (C macro), 5
USB_CDC_REQ_SET_CONTROL_LINE_STATE (C
macro), 5
USB_CDC_REQ_SET_LINE_CODING (C macro), 5
USB_CDC_SPACE_PARITY (C macro), 6
usbreset_isr (C++ function), 8
UVC_GET_CUR (C macro), 11
UVC_GET_MAX (C macro), 11
UVC_GET_MIN (C macro), 11

V

VA_EXPAND (C macro), 8
VA_NARGS (C macro), 8
VA_NARGS_INTERNAL (C macro), 8
valuesArray (C++ member), 11

W

WARN_UNUSED_RESULT (C macro), 8
WORD_LENGTH (C macro), 8