

# Fx2lib Introduction

November 19, 2015

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Fx2lib Features</b>	<b>1</b>
<b>3</b>	<b>Fx2lib Utilities</b>	<b>2</b>
<b>4</b>	<b>Getting Started</b>	<b>2</b>
<b>5</b>	<b>Where Next</b>	<b>3</b>

## 1 Introduction

Fx2lib is an open source library for the Cypress EZ-USB Fx2 8051 and variants. It is written in C and can be compiled with SDCC. Fx2lib contains libraries for performing common tasks with the Fx2 chip as well as a framework for starting new firmware projects. There are also a number of utilities for performing common firmware development tasks.

## 2 Fx2lib Features

Fx2lib provides libraries for the following common firmware tasks:

- Serial IO. Ease of setting BAUD rate and reading and writing from serial ports.

- Vendor Commands. Handling of required USB vendor commands and ease of implementing your own vendor command handlers.
- USB interrupts. Ease of enabling/disabling USB interrupts and adding your own interrupt handlers.
- I<sup>2</sup>C IO. Ease of reading and writing to I<sup>2</sup>C devices. EEprom functions are also included.
- GPIF. Functions for setting up the GPIF as well as performing single and fifo reads/writes.
- Endpoints. Simplify reading/writing with endpoint 0.
- Other definitions and macros. All Fx2 SFR, sbit and registers defined as well as other macros for common fx2 tasks.

Detailed documentation for the library API is available online or with your downloaded fx2lib copy.

### 3 Fx2lib Utilities

- gpif2dat. Takes the output from the Cypress GPIF designer (the .c file) and creates an sdcc compatible c file with only the waveforms and initialization structures needed for use with the Fx2lib GPIF functions.
- ihx2iic. Creates a compressed iic from your firmware ihx file. Used for storing your firmware on an eeprom.
- fx2load. Command line interface and Python bindings for loading firmware onto an Fx2 device and for doing basic device IO.

## 4 Getting Started

1. Obtain Fx2lib.

Fx2lib can be compiled from sources or is available as a pre-compiled library. If you choose to compile Fx2lib from sources, simply make sure sdcc is in your path and run make in the fx2lib root folder. Whether you have downloaded a pre-compiled version or build your own Fx2lib

library, you should have `fx2lib/lib/fx2.lib` available for the examples and your own firmware.

## 2. Install the drivers.

Fx2lib uses libusb to handle low level USB communication with the underlying operating system. Each operating system has different requirements for driver installation. See the Fx2lib driver guide for details on installing USB drivers for your specific operating system. You'll need to install the drivers specifically for the device you plan on doing the Fx2 development with.

## 3. Running Example Firmware.

Each firmware example has a Makefile and a load script. Simply run `make` to create the associated `.ihx` file and then run `make load` to load the firmware onto your device. The `examples/NOTES` contains more details on each examples firmware.

The default load targets use the Cypress vendor id with the development board product ID. Depending on your hardware, you'll need to change these appropriately.

## 4. Starting your own firmware.

Fx2lib provides a quick start firmware framework in the `fx2lib/fw` directory. Simply copy this directory to your own location and begin modifying the files to start your own firmware.

# 5 Where Next

- Fx2lib API documentation