

How to get USB 2.0 On The Go in minutes

The USB standard is one of the great success stories of recent years. High data rates and no-nonsense plug and play capability have made it the obvious choice for PC peripherals. It has replaced the once standard PC's serial and parallel interfaces. The user convenience we've come to expect of course comes at the cost of increased software and hardware complexity for designers.

In the world of embedded microcontrollers, the trusty UART by and large has remained the serial communications channel of choice between systems where USB compatibility is not needed. In order to address the market for USB compatible systems, many microcontrollers have appeared with USB protocol controllers integrated on chip. However, most only support USB as a peripheral device and rely on an external PHY for operation. Most of these microcontrollers also provide some software support for the USB peripheral, although it is typically not easy to use or easy to integrate with existing projects. Thus, a large amount of engineering time is spent becoming familiar with the hardware and software, and then implementing it within a system to achieve the project specifications.

The USB 2.0 OTG peripheral in Cyan's eCOG1X delivers a high level of functionality. However, the resulting level of complexity you'd expect is avoided by exploiting the CyanIDE configuration tool. This provides graphical configuration of the USB peripheral hardware and the low-level driver software. Additionally, high level libraries are selectable and automatically imported into the user's project. The configuration tool and libraries are provided free of charge, and cover many of the typical embedded USB applications including mass storage devices, sound cards, keyboards, pointing devices and game controllers. Additionally, the nature of the embedded peripheral allows easy bridging to any of the other eCOG1X microcontroller peripherals, vastly improving the design cycle for a range of USB powered devices. Currently applications support includes Ethernet bridging, camera modules, analogue I/O and RF transceivers. The result - easily integrated, expandable and re-usable firmware that allows USB applications to be built in minutes.

What's involved with OTG?

Going beyond the need for semi-intelligent PC peripherals, the USB core on the eCOG1X series supports peripheral mode, host mode and On The Go (OTG) operation. OTG is the extension to the USB 2.0 specification that allows a system to behave as either a host or a peripheral device, without requiring independent hardware interfaces to discriminate between the two. An OTG device utilises the micro-AB socket type, which allows the connection of both the micro-A and micro-B plug. The eCOG1X can detect whether it is connected as a host or peripheral device, depending on which plug is inserted – the ID pin is grounded for host or left floating for peripheral. The operation is dynamic, and subsequent re-configuration can not only be initialised by plugging in the relevant connector type but it may be switched back and forth under software control as the peripheral requests to become the host. OTG has readily found use in PDAs and mobile phones but there is also interest in industrial applications, such as data-logging and inventory control systems. The major benefits apply where a handheld or mobile unit connects at different times to a host in one place and peripheral units such as printers, flash disks and remote data monitors. In this scenario, one single standard interface connection can be used for all functions, removing the need for multiple hardware interfaces.

The USB peripheral in detail

An internal PHY is integrated on chip that supports Low Speed ("LS" - 1.5Mbps) and Full Speed ("FS" - 12Mbps) operation. Additionally, a ULPI interface is provided for use with an external PHY if High Speed ("HS" - 480Mbps) operation is required; ULPI is a standard 12 wire bus that will interface to a number of ULPI PHY devices.

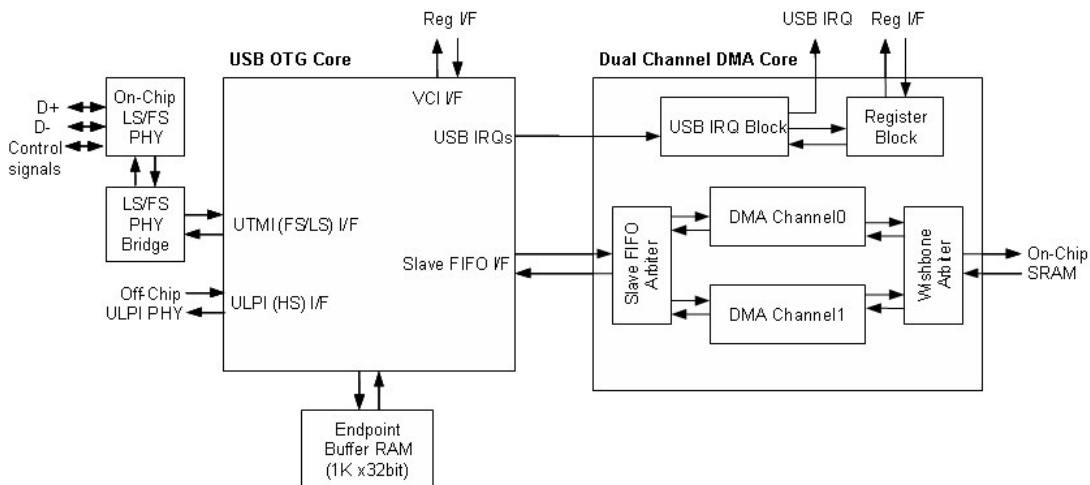


Figure 1. eCOG1X USB peripheral block diagram

Eliminating the software headache

Cyan's embedded software support separates the user application from the USB protocol and software stack using a well defined API. This enables the use of USB, without the need to understand the details. Applications can be created using the USB library at the start, or USB functionality can easily be added to existing programs.

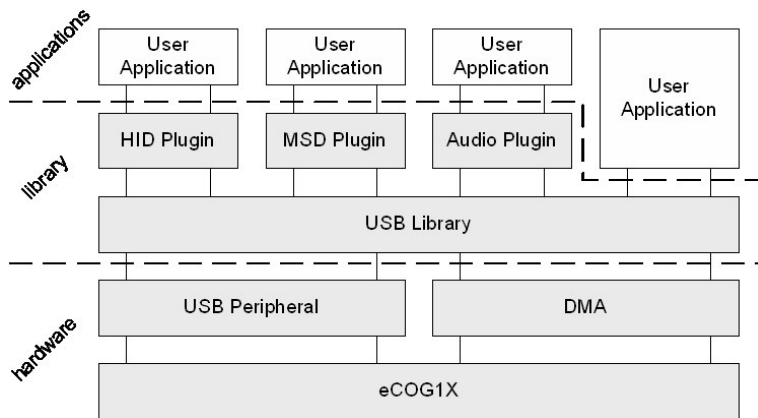


Figure 3. eCOG1X USB Library Hierarchy.

Hardware drivers are provided for operation of the USB core, FIFO and DMA access. Additionally, USB libraries are provided for accessing the low level functions, and support HID, MSD and Audio devices. The user application accesses the USB library and plug-ins directly, using a well defined API.

Library support is provided on a number of different levels, and can be divided up into the following functional areas:

- Setup
- Data transfer handlers – for FIFO and DMA buffers
- Interrupt handlers
- Plug-ins which provide device functionality with only configuration needed

Plug in drivers are available for the following USB Device Classes:

Human Interface Device (HID) - keyboard (host and peripheral)

HID devices use USB interrupt transfers, which are ideal for sending small amounts of information infrequently across the USB bus. The eCOG1X can be set to perform background processing and the USB transfers take very little processing time away from the main CPU. The HID class can be used for a variety of uses, including keyboard, barcode scanner (which is normally just a type of keyboard), mouse and joystick. HID is ideal for monitoring analogue or digital signals periodically.

Mass Storage Device (MSD) - read only disk (peripheral)

The eCOG1X simulates a large virtual mass storage device, potentially much larger than storage available to the microcontroller itself. This is useful for transferring large amount of data reliably from the eCOG1X to the host, taking advantage of the USB protocol's bulk transfer CRC error checking. For example, the microcontroller can present to the host as a virtual disk containing a large image file. When the host requests the file, an image will be grabbed from a connected camera and sent as the data. The next time the file is read, a new image is captured and sent. The eCOG1X USB core provides a level of control over the PHY, which allows it to perform a disconnect-reconnect under software control, forcing the host to re-read the device.

Mass Storage Device - flash memory disk (host) with FAT

Reading a connected USB flash drive is a common request, so the eCOG1X will interrogate a connected USB drive using the common SCSI-over-USB protocol. Low level block access is provided to read and write disk blocks of any disk format. Cyan also provides a general purpose FAT library, which provides file level read/write access to FAT16 and FAT32 file systems. The FAT library supports multiple partitions, multiple open files, standard C library style fopen(), fread(), fwrite() etc... routines.

Audio – external soundcard / headset (peripheral)

Rounding off the USB transfer types is isochronous transfer, which is demonstrated by the audio plugin. The audio plugin allows the eCOG1X to behave as an audio soundcard with line in and out at user set sample rate and sample size. Isochronous transfers provide realtime data transfers, but without CRC error checking. As well as audio transfer, video transfer is another popular use for isochronous transfers - this is the technique often used by webcams or digital cameras with realtime video preview facilities.

Under the CyanIDE environment, the USB core is included by drag and dropping it into the footprint image of the chosen chip variant. The software library supporting the chosen device is automatically added to the project, leaving very little work for the user. User support documentation is also available for the USB core drivers and all of the peripheral drivers and libraries via the CyanIDE help window. The documentation also includes a description of the user API to allow fast firmware prototyping.

Fig. 4 shows some example software that illustrates how simple it is to obtain the directory contents of a USB disk that's plugged into the eCOG1X, when using the CyanIDE MSD host library with FAT16/32 support. CyanIDE adds the plug-in for USB mass storage host and the FAT library provides all the routines necessary to read and write to the disk.

Figure 4. The USB peripheral code required to connect an MSD device.

```
msd_connect()
{
    int index, numDirectoryEntries;

    if (fat_mount(0)) {
        numDirectoryEntries = fat_get_num_dir_entries(NULL);

        for (index = 0; index < numDirectoryEntries; index++) {
            fat_print_dir_entry(fat_get_dir_entry(NULL, index));
        }
    }
}
```

The software library removes much of the required complexity from user applications, making configuration and subsequent connection a simple and efficient task

Overall the eCOG1X USB core offers a high level of functionality, covering peripheral, host and On The Go functionality. The hardware block is CPU autonomous, and incorporates fast and efficient DMA access to internal endpoint buffers. These features result in a large reduction in the amount of code needed to operate the device. A growing library of support software for numerous USB device classes and common peripherals is also provided at no charge. These libraries distance the user from the somewhat complex low level USB stack required to operate the chosen device, and replace it with high level routines designed for easy understanding and integration. Additionally, a wide choice of Cyan microcontrollers are available offering USB support, allowing the user to select the price points, memory and package sizes that suit any application.

To download free USB applications notes - 'USB hardware configurations', 'USB audio input and output' and 'USB FAT file system' - please visit http://www.cyantechnology.com/support/app_notes.php or contact sales@cyantechnology.com.