



CY4632 Protocol Library

Introduction

WirelessUSB™ LS is the ideal solution for a low-cost, low-power wireless device. The purpose of the CY4632 Protocol Library is to allow customers to leverage the WirelessUSB LS HID protocol to quickly develop WirelessUSB LS products. This document gives an overview of the CY4632 Protocol Library.

The CY4632 Protocol Library supports the following features:

- 64 kbps data rate (realized throughputs is 62.5kbps)
- 2-way communication
- 2:1 system configuration
- Automatic bind procedure

Background Information

This document assumes the reader to be familiar with the general operation of WirelessUSB LS. For more information on WirelessUSB LS please refer to WirelessUSB LS Theory of Operation, WirelessUSB LS 1-Way HID Systems, and WirelessUSB LS 2-Way HID Systems application notes.

WirelessUSB LS Peripherals

Peripheral Firmware Details

The CY4632 Protocol Library runs on a Cypress MicroSystems PSoC™ (CY8C26643-24PI) chip on the WirelessUSB KBM RDK boards (PDC-9166 and PDC-9174). The firmware is written in C for maximum portability to other platforms. The rest of this section gives a functional overview of the platform firmware.

File Descriptions

The CY4632 Protocol Library is split into two sections: Core Protocol files that contain protocol and radio interface code that is highly portable and Utility files that support the Core Protocol files and provide hooks into the environment. The Utility files will require porting if a non-PSoC environment is used. The purpose of the files in each directory is shown in the following table:

Filename	Purpose
Protocol Core	
<i>radio.c</i>	Collection of functions to access and control the LS Radio
<i>radio.h</i>	Include file which defines the LS registers and C function prototypes
<i>bind.c</i>	Collection of basic, semi-automatic, and automatic bind functions
<i>bind.h</i>	Include file which defines the bind prototypes and macros
<i>protocol.c</i>	Collection of function to perform the one-way and two-way protocols
<i>protocol.h</i>	Include file which defines the LS radio packet format, prototypes, and macros
Utilities	
<i>cypdef.h</i>	Include file which defines the system-wide typedefs
<i>nvrw.c</i>	Collection of functions to perform NVRAM read and write
<i>nvrw.h</i>	Include file which defines the NVRAM prototypes and macros
<i>spi.c</i>	Collection of SPI functions and macros
<i>spi.h</i>	Include file which defines the SPI prototypes and macros
<i>ls_config.h</i>	Main include file which determines the application, protocol, and radio parameters



Adapting the Peripheral Firmware

All protocol configuration options are contained in *ls_config.h*, allowing easy modification of the protocol and device configurations. To add a new device configuration the new header file must be included in *ls_config.h*. The header file must define TX_PERIOD, DEFAULT_PN_CODE, and the APP_TX_PACKET structure. Current protocol configuration options are described below.

DATA_RATE

This specifies the data rate to use over the air as described in WirelessUSB LS 1-Way HID Systems and WirelessUSB LS 2-Way HID Systems. The following data rates are available: DATA_RATE_16, DATA_RATE_32 and DATA_RATE_64. Data rates are given in kbps. One (and only one) data rate should be defined.

BIND_BASIC

This specifies the basic bind procedure, used in both 1-way and 2-way protocols.

BIND_SEMI

This specifies the semi-automatic bind procedure, used in the 1-way protocol.

BIND_AUTO

This specifies the automatic bind procedure, used in 2-way protocol.

TWO_WAY

This specifies the 2-way protocol. If this is not defined the 1-way protocol is used.

BACK_CHANNEL_DATA

This enables back channel data to be received from the bridge or another WirelessUSB LS device as described in WirelessUSB LS 2-Way HID Systems. If ASYNC_BACK_CHANNEL_DATA is not defined (see below) only polled back channel data will be received.

DEFAULT_PN_CODE

This is the index of the PN Code to be used. Basic bind never changes the PN Code after the DEFAULT_PN_CODE is loaded during initialization; semi-automatic and automatic bind procedures can dynamically change the PN Code after the DEFAULT_PN_CODE is loaded during initialization. The DEFAULT_PN_CODE can be any value between 0 and 47 for DATA_RATE_16 devices, between 0 and 7 for DATA_RATE_32 or DATA_RATE_64 devices.

PA_BIAS

This controls the transmit power of the radio. The default is 07. (Range: 0-7)

DYNAMIC_PA_SUPPORT

This enables support for dynamic control of the transmit power on the radio.

APP_TX_PACKET_SIZE

This is the maximum size the application transmit packet can be. Variable length packets can be used; see the PERT and General applications for sample code.

APP_RX_PACKET_SIZE

This is the maximum size the application receive packet can be. Variable length packets can be used; see the PERT and General applications for sample code.

SLEEP_ENABLED

This enables code in the radio logic to put the radio to sleep when not in transmit or receive modes.

Peripheral Firmware Details

The CY4632 Protocol Library makes extensive use of "C" macros to determine the mode of operation. The details of the protocol and radio are hidden behind the API functions. The application makes a standard API call and the appropriate code is called with the use of the compile time macro options. The application logic doesn't change because of the bind mode selected. The Utility files are the only files that need to be ported if another CPU and/or board are to be used.



Significant API Functions

`protocol_init()`

This routine initializes the protocol logic and the radio hardware. It should be called before any other protocol functions are called.

`protocol_get_tx_pkt()`

This routine returns a pointer to a buffer of length `APP_TX_PACKET_SIZE`. The application loads data into the buffer and then calls `protocol_send_packet()`. This function only needs to be called once prior to the first time the application calls `protocol_send_packet()`.

`protocol_bind()`

This routine performs the automatic bind procedure as described in *WirelessUSB LS 2-Way HID Systems*.

`protocol_send_packet()`

This routine formats and sends the application's data packet to the radio and then handles the 2-way logic if selected.

`protocol_get_packet()`

This routine gets packets from the radio ISR, and then handles the 2-way logic if selected, and makes the data available to the application.

`radio_rx_isr()`

The CY4632 Protocol Library polls the radio interrupt in the radio transmit logic. The radio interrupt is used in the radio receive logic. When a byte is received in the radio it toggles the IRQ pin, which should trigger an I/O interrupt on the microprocessor, which in turn calls which then calls `radio_rx_isr`. This function stores the byte in a packet buffer. When an end of frame occurs the protocol library will process the received packet.

References

- *WirelessUSB LS Theory of Operation Whitepaper*
- *WirelessUSB LS 2-Way HID Systems*
- *WirelessUSB LS Development Kit User's Guide*
- *Device Class Definition for Human Interface Devices (HID)* (<http://www.usb.org/developers/hidpage>)
- *CYWUSB6934 Datasheet (38-16007)*

WirelessUSB and enCoRe are trademarks of Cypress Semiconductor.

PSoC is a trademark of Cypress MicroSystems, a subsidiary of Cypress Semiconductor.