Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

THIS SPEC IS OBSOLETE

Spec No: 001-83281

Spec Title: AN83281 - DEVELOPING RF-BASED REMOTE CONTROL USING WIRELESSUSB(TM)-NL

Replaced by: None

AN83281

Developing RF-Based Remote Control Using WirelessUSB™-NL

Author: Dikshak Pandya
Associated Project: Yes
Associated Part Family: CY3668 WirelessUSB™-NL DVK
Software Version: PSoC® Designer™ 5.3 or higher
Related Documents: For a complete list of the application notes, click here.

AN83281 shows you how to use Cypress's WirelessUSB™-NL radio technology to develop an RF-based remote control for smart TVs and set-top boxes. The resulting application overcomes the major shortcomings of IR-based remote control. Included in the application note are descriptions of the system architecture and associated functional modules and PSoC projects.

Contents

1 Introduction....1
2 WirelessUSB Resources....3
2.1 PSoC Designer....3
3 WirelessUSB Remote Control System Architecture....4
3.1 WirelessUSB Dongle....5
3.2 WirelessUSB Remote Control Device....8
4 Remote Control Functional Demonstration Using Code Example....10
4.1 System Requirements....10
4.2 PSoC Designer Project for WirelessUSB Dongle....10
4.3 PSoC Designer Project for WirelessUSB Remote Control....17
4.4 CY3668 DVK Board Setup for WirelessUSB Remote Control....18
4.5 Remote Control Functional Operation....18
5 Summary....19
6 Related Documents....20
7 Appendix A. WirelessUSB Dongle Application Flow Chart....21
8 Appendix B. WirelessUSB Remote Control Flow Chart....22
9 Appendix C. Message Sequence Chart for WirelessUSB Remote Control Transactions....23
Worldwide Sales and Design Support....25

1 Introduction

This application note focuses on an RF-based remote control device operating over the 2.4-GHz ISM (industrial, scientific, and medical) band. Because it uses Cypress's proprietary WirelessUSB-NL radio, the device overcomes the following shortcomings of IR-based remote control:

- line-of-sight pointing
- limited operating angles
- short transmission range
- reflection problems
- high current consumption

The remote control device consists of a WirelessUSB Bridge/Dongle and a WirelessUSB Remote Control. Both devices use Cypress's low-power 2.4-GHz WirelessUSB-NL radio (CYRF8935).

A WirelessUSB Dongle is attached to a USB HID (human interface device)-compliant smart TV, a digital set-top box (STB) or a PC. The dongle enumerates as a composite USB Device with an HID-compliant consumer control interface, keyboard and HID mouse interfaces. In addition, the dongle adds WirelessUSB Host transceiver capability to a USB HID-compliant smart TV or a set-top box to communicate with a WirelessUSB Remote Control device and other WirelessUSB HID devices. (The terms WirelessUSB Bridge and WirelessUSB Dongle are used interchangeably in this application note.)

A WirelessUSB Remote Control and Dongle use the proprietary Enhanced AgileHID™ protocol for wireless communication.

The Enhanced AgileHID is a proprietary protocol designed for low-power wireless HID applications, such as a mouse, keyboard, consumer control device, and presenter. The protocol works with WirelessUSB technology using WirelessUSB-NL radio.

The WirelessUSB Dongle application runs on an enCoRe™ V (Enhanced Component Reduction) 8-bit microcontroller, supporting a full-speed USB controller, and the WirelessUSB Remote Control application runs on an enCoRe™ V LV microcontroller. Refer to *enCoRe™ V LV TRM* and *WirelessUSB™-NL Radio TRM* for complete details.

The code example application in this application note demonstrates the remote control functionality using a PC. (A USB-HID-compliant smart TV was not available.) The WirelessUSB Remote Control application can control media player application functions, such as Play/Pause, Stop, and Volume Control, on the USB HID-compliant smart TV or a PC. Here's how the remote control application works:

- The WirelessUSB Remote Control establishes the wireless link with the WirelessUSB Dongle attached to the smart TV or PC.
- It sends WirelessUSB data packets for USB HID-compliant remote control reports.
- The WirelessUSB Dongle converts the data packets to the USB HID-compliant consumer control report descriptors.
- It sends the data packets to the USB HID-compliant application running on a smart TV or PC.

  You can add functions and user-defined smart features to enhance the attached example code. For more information, see the USB HID usage table specification, HUT 1.12.

Figure 1. WirelessUSB Remote Control System

2 WirelessUSB Resources

Cypress provides a wealth of data at www.cypress.com to help you to select the right WirelessUSB device for your design, and quickly and effectively integrate the device into your design. For a comprehensive list of resources, see the wireless webpage.

- **Overview:** Wireless Roadmap , Modules Roadmap , Wireless Portfolio
- **Product Selectors:** Wireless Product selector
- **Datasheets** describe and provide electrical specifications for various device families. You can access the datasheet of all wireless products here.
- **Application Notes and Code Examples** cover a broad range of topics, from basic to advanced level. Many of the application notes include code examples. You can access the complete list of wireless AN here.
- **Technical Reference Manuals (TRM)** provide detailed descriptions of the architecture and registers in each WirelessUSB device family.

  You can access the complete list of wireless products TRM here.
- **Development Kits**:

  You can access the complete list of wireless kits and reference designs here

Cypress also offers ARM Cortex-M0 based, single-chip Bluetooth Low Energy (BLE) or Bluetooth Smart solutions. You can learn more about Cypress BLE devices here.

Cypress has a BLE based remote control RDK. You can get more details about the RDK here.

2.1 PSoC Designer

PSoC Designer is the revolutionary Integrated Design Environment (IDE) that you can use to customize PSoC to meet your specific application requirements. PSoC Designer software accelerates system bring-up and time-to-market. Develop your applications using a library of pre-characterized analog and digital peripherals in a drag-and-drop design environment. Then, customize your design leveraging the dynamically generated API libraries of code. Finally, debug and test your designs with the integrated debug environment including in-circuit emulation and standard software debug features.

- Application Editor GUI for device and User Module configuration and dynamic reconfiguration
- Extensive User Module Catalog
- Integrated source code editor (C and Assembly)
- Free C compiler with no size restrictions or time limits
- Built-in Debugger
- Integrated Circuit Emulation (ICE)
- Built-in Support for Communication Interfaces:
  - Hardware and software I2C slaves and masters
  - Low/Full-speed USB 2.0
  - Up to 4 full-duplex UARTs, SPI master and slave, and Wireless

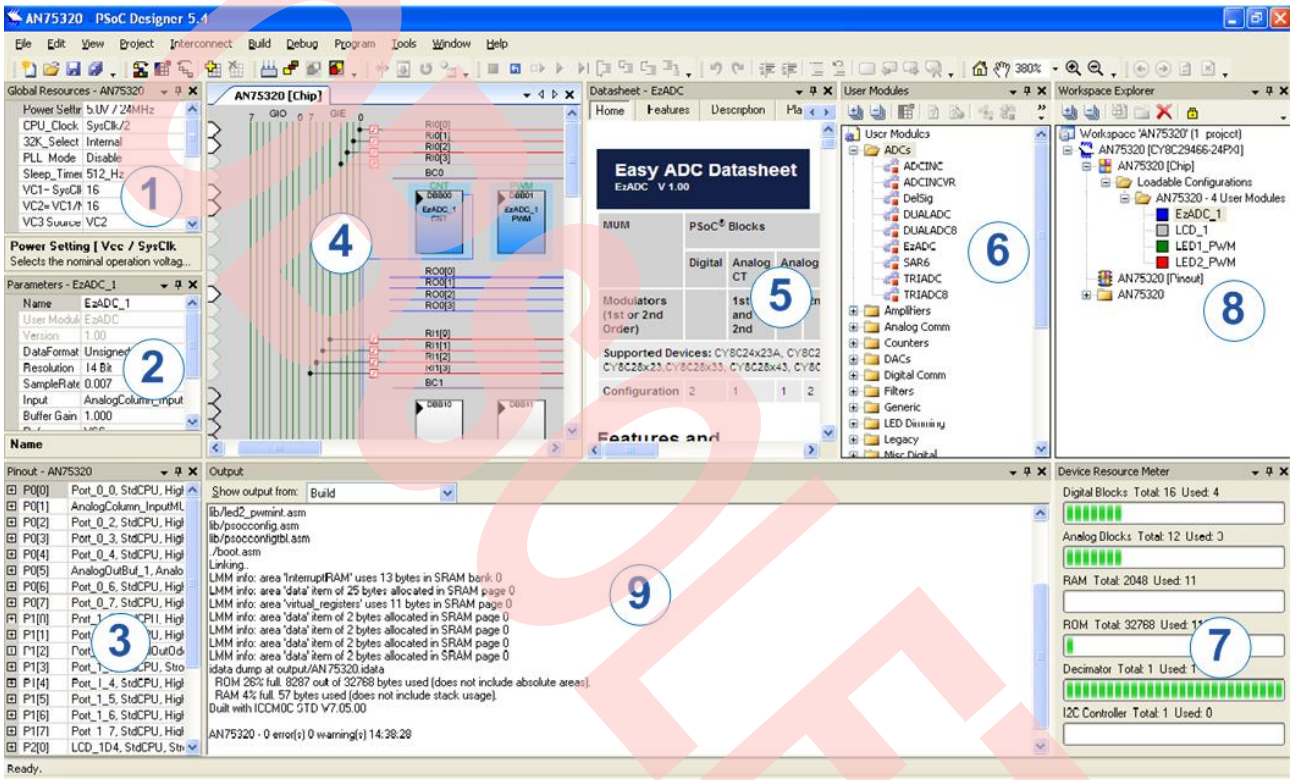
Figure 2 shows PSoC Designer windows. **Note:** This is not the default view.

1. **Global Resources –** all device hardware settings.
2. **Parameters –** the parameters of the currently selected User Modules.
3. **Pinout –** information related to device pins.
4. **Chip-Level Editor –** a diagram of the resources available on the selected chip.
5. **Datasheet –** the datasheet for the currently selected UM
6. **User Modules –** all available User Modules for the selected device.

7. **Device Resource Meter –** device resource usage for the current project configuration.
8. **Workspace –** a tree level diagram of files associated with the project.
9. **Output –** output from project build and debug operations.

Note: For detailed information on PSoC Designer, go to **PSoC® Designer** > **Help** > **Documentation** > **Designer Specific Documents** > **IDE User Guide**.

Figure 2. PSoC Designer Layout



3 WirelessUSB Remote Control System Architecture

The WirelessUSB Remote Control application is demonstrated using the enCoRe V-based CY3668 WirelessUSB NL Development Kit with the low-power WirelessUSB-NL radio transceiver. Use Cypress's PSoC Designer™ Integrated Development Environment (IDE) to create WirelessUSB Remote Control and WirelessUSB Dongle applications.

PSoC Designer is a hardware and software co-design environment based on schematic entry and embedded design methodology. The IDE simplifies the configuration and the interconnection of controller peripheral devices, such as an interrupt controller, timers, GPIO, and USB.

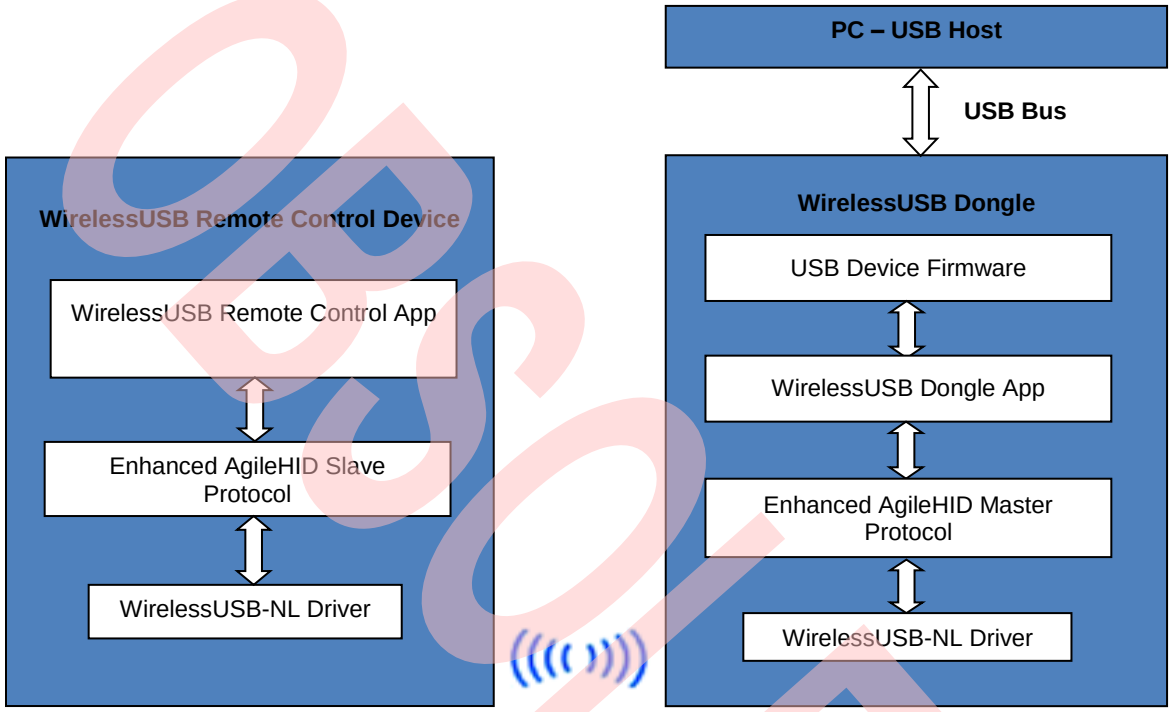
Using PSoC Designer, you can automate code generation for User Modules of controller peripherals based on their configurations.

The WirelessUSB Remote Control system is composed of a WirelessUSB Dongle and WirelessUSB Remote Control device. Figure 3 shows the layered architecture of the system.

The WirelessUSB Dongle functions as the WirelessUSB Host device and manages all of the WirelessUSB Devices connected to it. It uses the Master mode of the Enhanced AgileHID protocol to interact with the underlying radio and to manage all the WirelessUSB communication. PSoC USB User Module library APIs communicate with a USB-HID-compliant application using the PC USB Host controller and USB-HID function driver.

The WirelessUSB Remote Control is a native WirelessUSB HID device that uses the slave mode of the Enhanced AgileHID protocol to communicate with the WirelessUSB Host. The remote control manages a push button-based user interface through GPIOs. You can enhance the example application for WirelessUSB Remote Control code to create an interface with other user inputs, such as touch or keys.

Figure 3. Remote Control System Layered Architecture



3.1 WirelessUSB Dongle

The WirelessUSB Dongle is implemented using the enCoRe V-based CY3668 DVK with WirelessUSB-NL radio. The enCoRe V USB User Module adheres to the USB2.0 specification for full-speed devices. Specifically, it supports control, interrupt, bulk, and isochronous transfers. For more details, refer to the enCoRe V USB User Module datasheet.

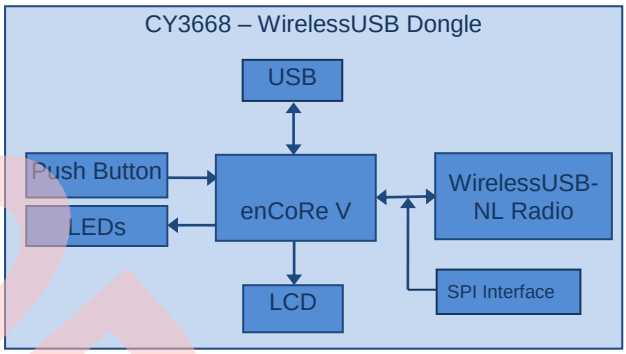
The WirelessUSB Dongle is composed of the following functional modules:

- PSoC USB Device Firmware
- WirelessUSB Dongle application
- Enhanced AgileHID Master Protocol
- WirelessUSB-NL Driver

The WirelessUSB Dongle application uses the PSoC library APIs for User Modules USB, GPIOs, SPI, and the Timer. Push buttons work with GPIOs as the user input for entering bind mode of the WirelessUSB Remote Control device. The Enhanced AgileHID Master protocol uses the Timer to control the operations of the WirelessUSB Device.

Figure 4 shows the block diagram of the WirelessUSB Dongle.

Figure 4. WirelessUSB Dongle Block Diagram



3.1.1 USB Device Firmware

This module is responsible for USB Device enumeration. It handles USB control and data transactions between the USB Host and a USB Device.

The WirelessUSB Dongle application uses APIs exposed by this module for USB User Module initialization, USB control, and data transactions. For a list of APIs, refer to the PSoC USB User Module datasheet.

The PSoC Designer USB User Module includes the USB setup wizard, which lets you modify the Device Descriptor, Configuration Descriptor, Interface Descriptor, and Report Descriptor to fit your application.

Enumeration Process

The enumeration process consists of requests made by the USB Host for descriptors stored in the USB Device. The descriptors describe the device's capabilities.

The device must respond with descriptors that follow a standard format according to the USB specification. Descriptors contain basic information about a device response to device descriptor, configuration, interface and report descriptor requests by the USB Host.

The WirelessUSB Remote Control application enumerates as a composite USB Device. It can support a USB HID consumer-compliant remote control device, a USB HID keyboard, and USB HID mouse interfaces.

An HID-class device identifies its data protocol and the type of data provided within its Report descriptor.

The Report descriptor is loaded and parsed by the HID class driver as soon as the device is detected. Protocols for existing and new devices are created by mixing data types within the Report descriptor.

Figure 5 shows a sample Report descriptor template for an HID consumer-compliant PC-based multimedia remote control device. It provides the following functions:

- Transport Control (Play/Pause, Stop)
- Audio Control (Vol. Increase, Vol. decrease)
- Figure Report descriptor for Remote Control Device

Figure 5. HID-Compliant Consumer Control Report Descriptor

Table 1. Report Descriptor Data for Remote Control

b7	b6	b5	b4	b3	b2	b1	b0
Pad bits				Vol. Down	Vol. Up	Stop	Play/Pause

The nature and size of USB HID data are determined by using the report descriptor of the HID class driver.

Table 1 shows the USB HID data for HID report descriptor for a remote control application with audio control and media transport control operation. You can group and construct application-specific functional controls and add programmable buttons for the consumer control page based on the HID usage table, per *HUT1.12 specification*.

The setup wizard of PSoC Designer's USB User Module has a sample report descriptor template, which lets you create and edit the report descriptor format to meet your requirements.

Refer to the *USB HID specification* for a detailed description of each HID report descriptor item. Based on the report descriptor, the data is formed and interpreted by the HID class driver.

You can add more consumer control features to the report descriptor, as Figure 4 shows. Refer to the HID usage table document for a detailed description of available HID-compliant consumer controls.

USB transactions

The WirelessUSB Dongle uses a control endpoint pipe to control data exchange in the form of descriptors with the USB Host, as follows:

- Receiving and responding to requests for USB control and class data
- Transmitting data when polled by the HID class driver for Get_Report requests
- Receiving data from the host

The WirelessUSB Dongle uses an **Interrupt** pipe for the following two functions:

- Sending asynchronous (unrequested) data received from the WirelessUSB HID device to the USB host device for consumer-compliant remote control, keyboard, and so on. Typical HID data can be "key presses," "switch data," or "motion sensor data."
- Transmitting low-latency data to the device

3.1.2 WirelessUSB Dongle Application

This module is responsible for the following functions:

- USB Device control function
  - USB Device initialization
  - USB Device power save mode
- WirelessUSB Host control function
  - Connect/Disconnect, Bind Procedure
  - Power save mode
  - Data transformation for WirelessUSB data packets to HID USB data

    The WirelessUSB Dongle application controls downstream WirelessUSB native wireless HID devices, such as wireless consumer-compliant remote control, wireless keyboard, and wireless mice using the Enhanced AgileHID protocol.

3.1.3 Enhanced AgileHID Master Protocol Layer

The Enhanced AgileHID protocol is configured in master mode during compile time for the WirelessUSB Dongle application. Enhanced AgileHID Master Protocol controls over-the-air wireless transactions with native WirelessUSB HID devices.

The Enhanced AgileHID protocol communicates with the radio Hardware Abstraction Layer (HAL) layer to control different types of low-power 2.4-GHz RF transceivers.

The Enhanced AgileHID Protocol provides a controlled interface in the form of APIs to the application layer for the following services:

- Radio configuration and initialization
- Bind process: Establish a physical connection between the WirelessUSB Dongle and the device-over-radio link
- Packet receive and packet transmit operation
- Back-channel data requests and responses
- Quality Of Service (QoS) management for application layer by means of channel change and flow control mechanisms
- USB Device power state management

Refer to the *Enhanced AgileHID™ API* document for more details.

3.1.4 WirelessUSB-NL Radio Driver

The radio driver library of the WirelessUSB-NL radio (CYRF8935) shows an API interface to configure, initialize, and control the radio operations for data transmission and data reception. The CYRF8935 chip communicates with the enCoRe V device through an SPI bus. It allows the upper layer to control the radio state machine. It also allows the upper layer to configure functional blocks of Framer and the digital modem for performance. Refer to *CYRF8935 datasheet* for complete details.

Here are the basic operations and features of the 2.4-GHz WirelessUSB-NL:

- Data transmission and reception
- Data whitening
- Flow control mechanism
- Data integrity mechanism using optional FEC and CRC, packet buffering
- Event indication for data reception to upper layer
- Event indication for transmission status

Appendix A. WirelessUSB Dongle Application Flow Chart captures the WirelessUSB Dongle flow chart.

3.2 WirelessUSB Remote Control Device

The WirelessUSB Remote Control device is a wireless remote control application built using Cypress's 2.4-GHz WirelessUSB technology. It is built on the CY3668 WirelessUSB™-NL DVK with an enCoRe V LV core. Figure 3 shows the functional diagram for WirelessUSB Remote Control.

WirelessUSB Remote Control is composed of the following modules:

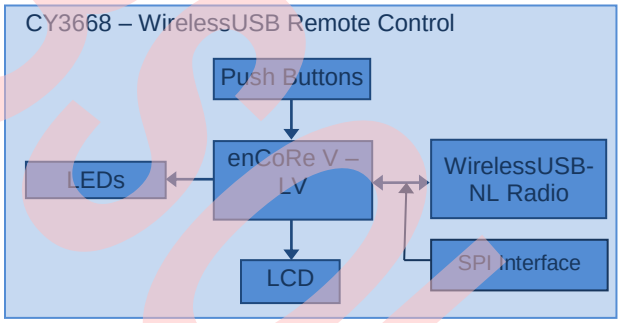
- Remote control application layer
- Enhanced AgileHID slave protocol layer
- WirelessUSB-NL radio driver

3.2.1 WirelessUSB Remote Control Application Layer

The WirelessUSB Remote Control application uses the Enhanced AgileHID slave protocol layer to communicate wirelessly with the WirelessUSB Dongle using 2.4-GHz WirelessUSB-NL transceiver module. Figure 6 shows the block diagram for WirelessUSB Remote Control.

The WirelessUSB Remote Control application uses PSoC User Module library APIs for GPIO, SPI, and Timer. It configures the GPIOs for push buttons as the user input for entering bind mode of the WirelessUSB Remote Control device and for adding consumer control functions. Timers are mainly used by the Enhanced AgileHID Slave protocol to control the operations of the WirelessUSB Device.

Figure 6. WirelessUSB Remote Control Block Diagram



The application layer also interacts with system peripherals to control HID device state and initiate control and data transactions.

The remote control application layer manages the following functions:

- System initialization
- Initiate and maintain wireless connection with WirelessUSB Dongle using Enhanced AgileHID Slave protocol layer services
- User input data transformation to appropriate HID data format
- Controls HID data transactions with the WirelessUSB Dongle device using Enhanced AgileHID Slave protocol layer services
- Maintains HID device's state information and state transitions

3.2.2 Enhanced AgileHID Slave Protocol Layer

Enhanced AgileHID protocol is configured in slave mode during compile time for the WirelessUSB Remote Control application. The Enhanced AgileHID slave protocol synchronizes and communicates with Enhanced AgileHID master Dongle for wireless transactions.

Enhanced AgileHID protocol communicates with the radio HAL layer to control different types of low-power, 2.4-GHz RF transceivers.

Enhanced AgileHID Slave Protocol provides the APIs to the application layer for the following services:

- Radio configuration and initialization
- Binding process: Establish physical connection between the WirelessUSB Device and the dongle-over-radio link
- Transmission and reception of WirelessUSB data over radio link
- Use channel change and flow control mechanisms to maintain QoS for application layer

- The WirelessUSB-NL Driver module is the same for both the WirelessUSB Dongle application and the WirelessUSB Remote Control device.

Appendix B. WirelessUSB Remote Control Flow Chart captures the WirelessUSB remote control flow chart.

4 Remote Control Functional Demonstration Using Code Example

4.1 System Requirements

- CY3668 DVK
- PC (Windows XP SP2, Vista, 7, 8)

You can demonstrate remote control functionality on a CY3668 DVK board setup using sample application firmware code attached with the application note.

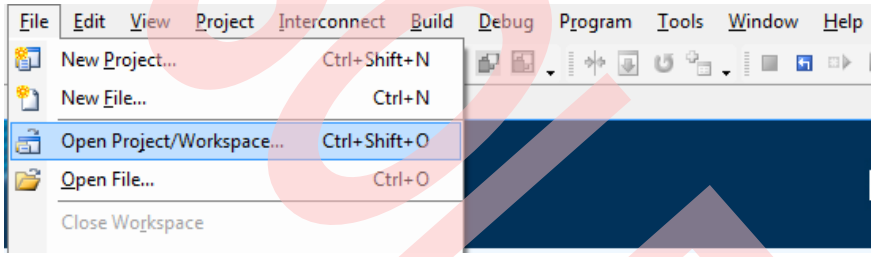
The following steps describe how to run the remote control application demonstration.

AN83281 has two associated code example projects: WirelessUSB Dongle and WirelessUSB Remote Control.

4.2 PSoC Designer Project for WirelessUSB Dongle

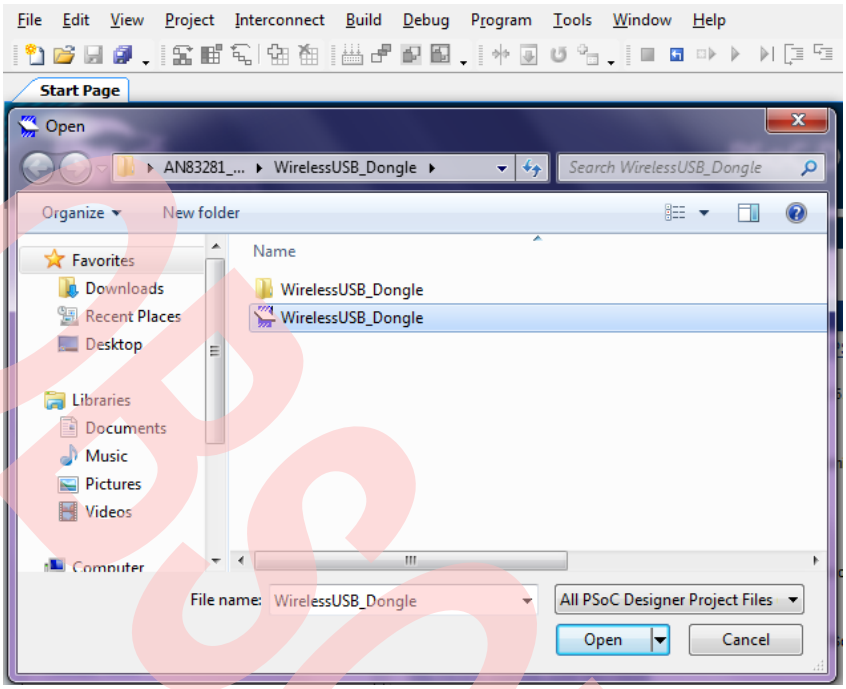
1. Run PSoC Designer 5.3. Open the project 'WirelessUSB-*NL_Dongle'* for dongle application in PSoC Designer, as Figure 7 shows.

Figure 7. PD 5.3 Open Project Dialogue Box



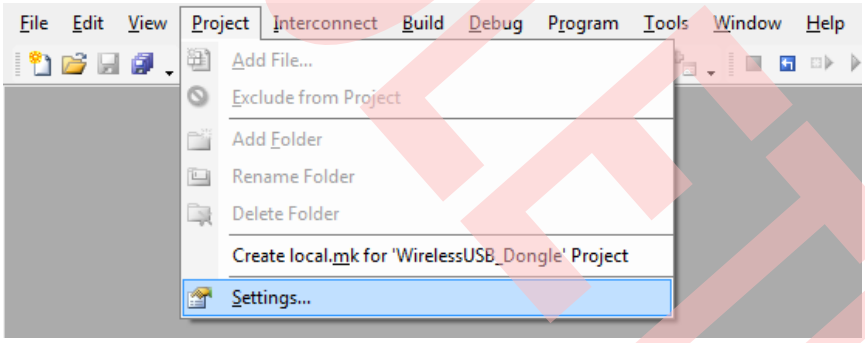
2. Select WirelessUSB_Dongle project from project browse dialogue box, as Figure 8. shows.

Figure 8. PD 5.3 Project Browse Dialogue Box



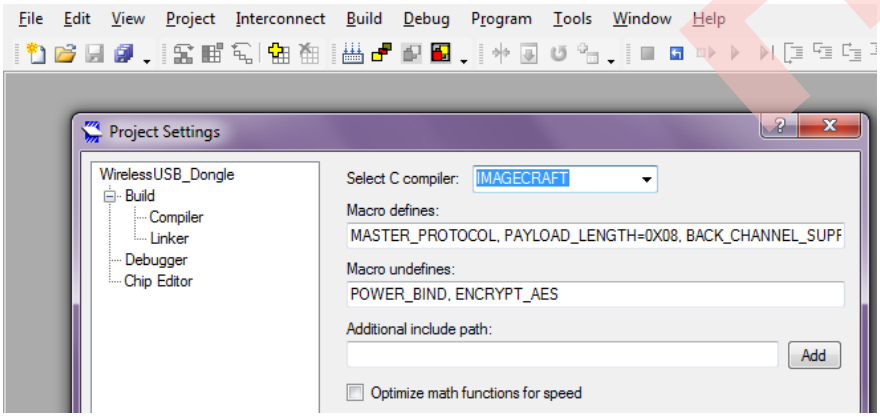
3. Go to **Project** > **Settings** > **Compiler**, as Figure 9 shows.

Figure 9. PD 5.3 Project Settings Traverse



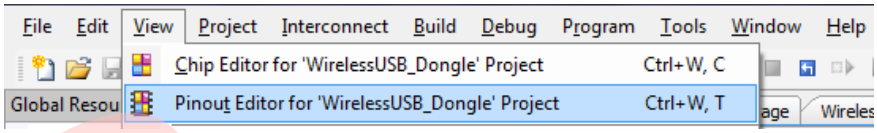
4. Select Image Craft as compiler option, as Figure 10 shows.

Figure 10. PD 5.3 Project Settings



5. Go to **View** > **Pinout** Editor, as Figure 11 shows.

Figure 11. PD 5.3 Project Pinout Editor

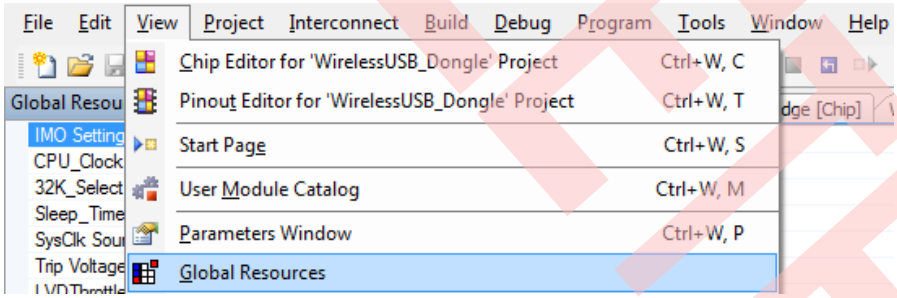


6. Now verify the pinout settings, as Figure 12 shows.

Figure 12. Pinout Settings for WirelessUSB Dongle

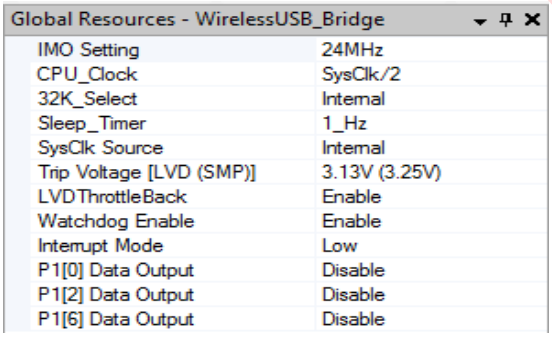
7. Go to **View** > **Global Resources** menu, as Figure 13 shows.

Figure 13. PD 5.3 Project Settings



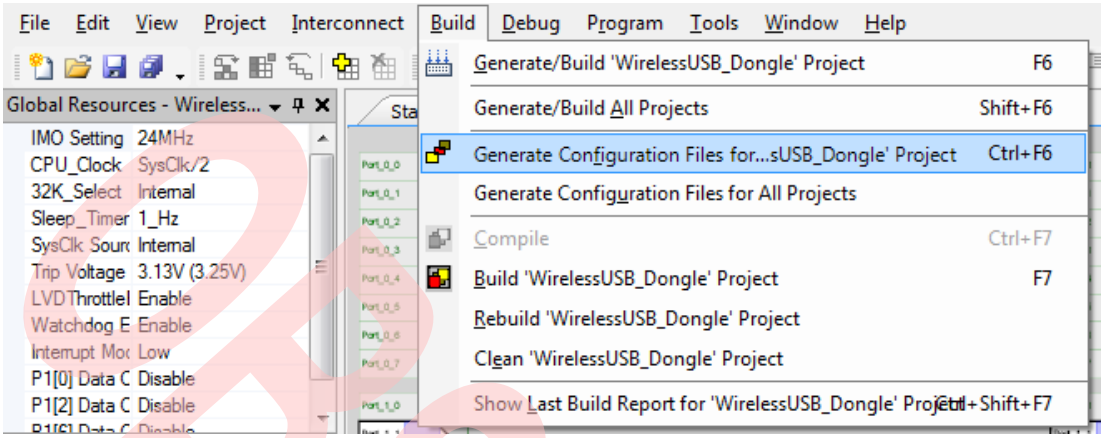
8. Verify the Global Resource Settings, as Figure 14 shows.

Figure 14. PD 5.3 Global Resource Settings for WirelessUSB Dongle



9. Select **Build** > **Generate Configuration Files** for ' *WirelessUSB-NL_Dongle'* Project, as Figure 15 shows.

Figure 15. PD 5.3 Project Build Menu

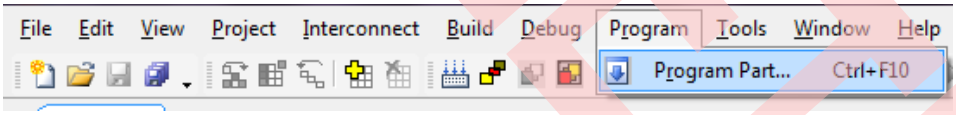


10. Build the project. Select **Build > Build** *'WirelessUSB-NL_Dongle'* Project or press **[F7]**.
11. For enCoRe V module, turn the SW1 switch downward on the CY3668 board; for enCoRe II module, turn SW1 upward.

4.2.1 Programming Steps

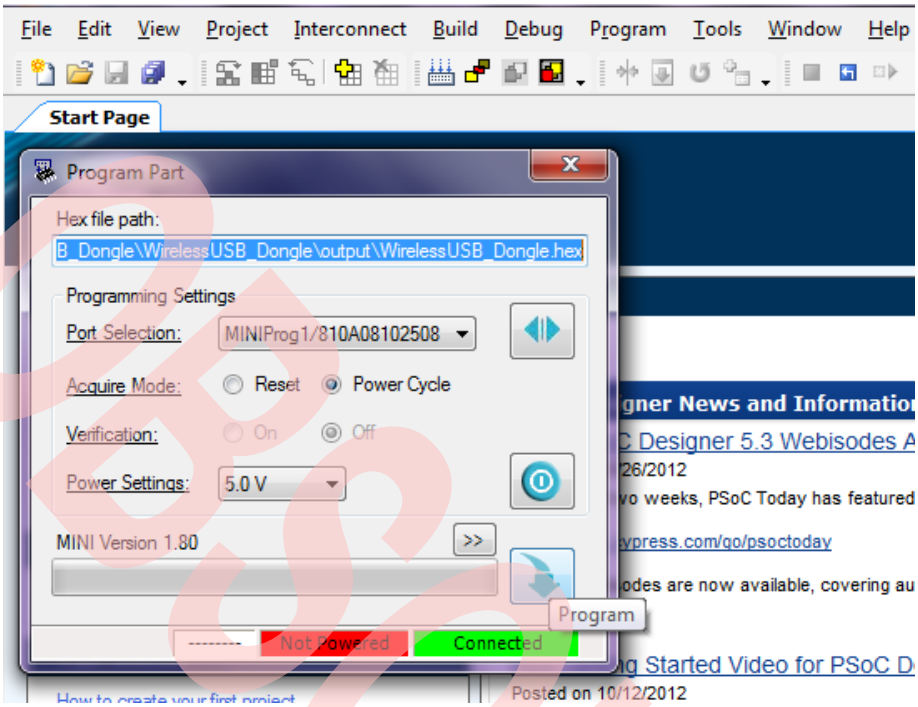
1. Check to ensure that WirelessUSB-NL Radio Module is not connected to the NL radio interface connector (P2) during programming enCoRe V on CY3668 board.
2. Connect one end of the USB cable to the PSoC MiniProg and the other end to the PC.
3. Connect the PSoC MiniProg to the 5-pin ISSP header on the CY3668 DVK board.
4. Program the CY3668 DVK board with the output hex file located at "*\WirelessUSB™-NL_Dongle\WirelessUSB™-NL_Dongle\output*".
5. Go to **Program** > **PSoC Programmer** menu in PSoC Designer, as Figure 16 shows.

Figure 16. PD 5.3 Program Menu



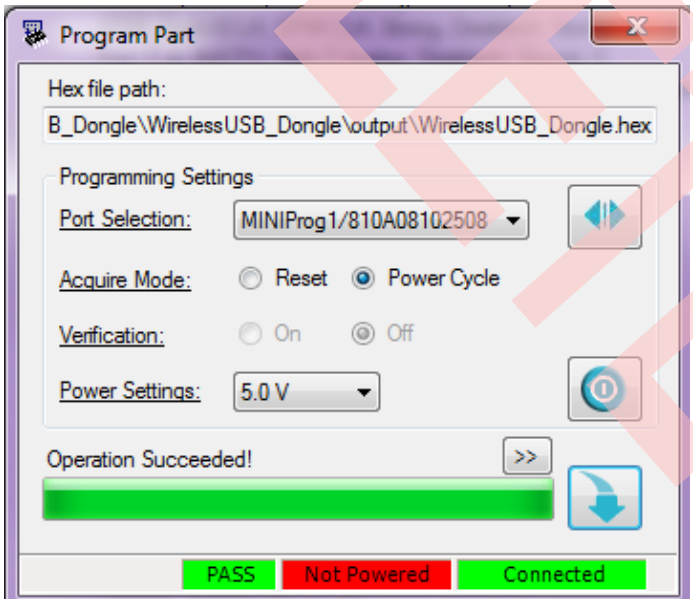
6. Verify the programming, as Figure 17 shows.

Figure 17. PD 5.3 Program Part Dialogue Box



7. Click the Program button to start programming the enCoRe V on the CY3668 board. Verify that the programming operation was successful, as Figure 18 shows.

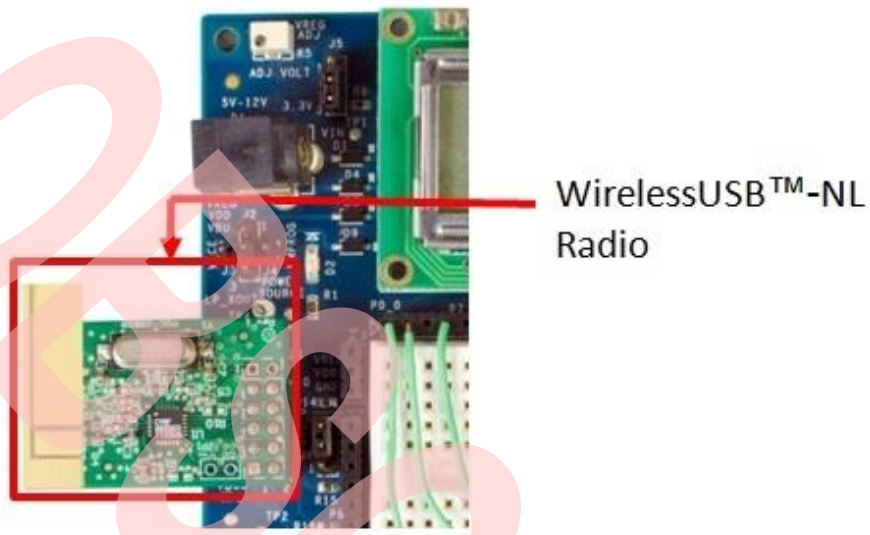
Figure 18. PD 5.3 Program Operation Status



4.2.2 CY3668 DVK Board Setup for WirelessUSB Dongle

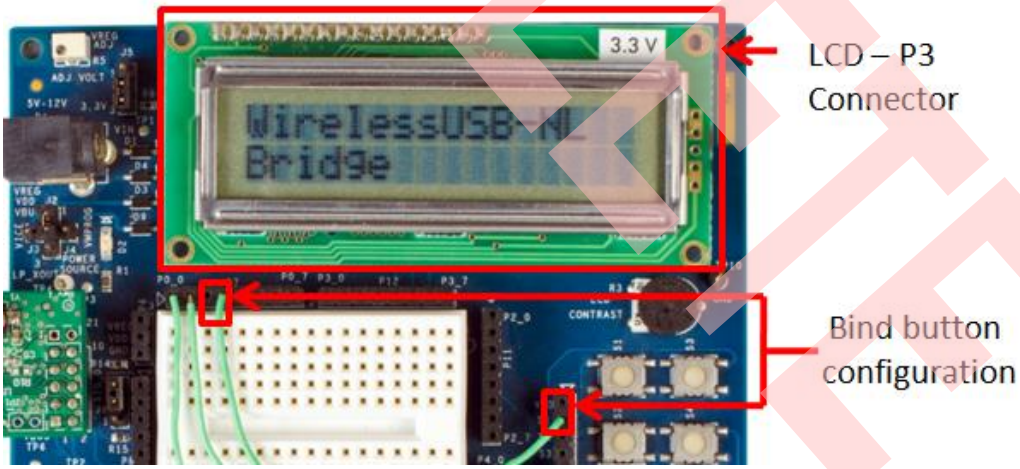
1. Attach the WirelessUSB-NL Radio module to P2, as Figure 19 shows.

Figure 19. WirelessUSB-NL P2 Connector



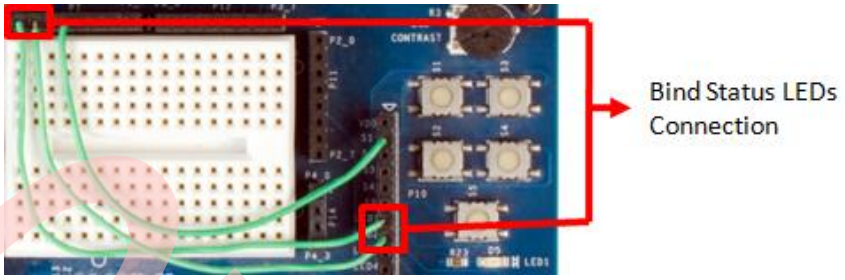
2. Mount the LCD module on the P3 connector of the CY3668 DVK board, as Figure 20 shows.
3. Configure Bind button: Wire up button S1 as Bind button, connect pin P0_3 on connector P7 to S1 on connector P10 with wire on CY3668 DVK board, as Figure 20 shows.

Figure 20. S1 Bind Button, LCD Configuration



4. Configure LEDs. Wire up LED1 and LED2, Connect pin P0_0 on connector P7 to LED1 on connector P10 and from pin P0_1 on connector P7 to LED2 on connector P10 on the CY3668 DVK board, as Figure 21 shows.

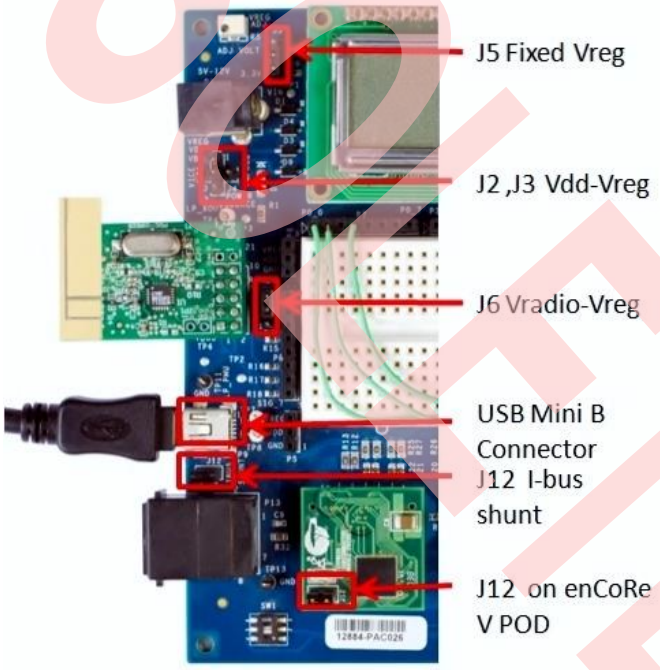
Figure 21. Bind Status LEDs Configuration



5. Configure Radio, Power:
   - Place a two-pin jumper on J12.
   - Place a two-pin jumper on J6 connecting VRADIO and VREG.
   - Place a two-pin jumper on J2 and J3, connecting VDD and VREG.
   - J5 is used to select the fixed 3.3-V VREG or adjustable VREG from 3.66 V to 1.63 V. In this example, we use fixed 3.3-V VREG.

   Jumper settings are shown in Figure 22.

Figure 22. Jumper Settings and USB Mini – B Connector

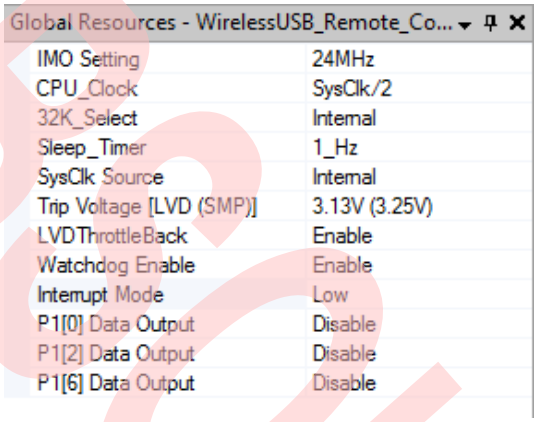


6. Configure PC – WirelessUSB Dongle Connection:
   - Connect the mini-B side of the USB A/Mini-B cable to the CY3668 DVK board.
   - Connect the A side of the USB A/Mini-B cable to the PC.
   - The device should enumerate on the PC as a composite USB Device supporting the HID-compliant consumer control device, HID keyboard, and HID mouse interfaces.

4.3 PSoC Designer Project for WirelessUSB Remote Control

1. Follow the steps with snapshots as described for WirelessUSB Dongle project to open, build and program the WirelessUSB Remote Control PSoC Project with PSoC Designer 5.3.
2. Open the PSoC Designer project '*WirelessUSB-NL_Remote_Control'* for remote application in PSoC Designer. This project is configured for the enCoRe V device.
3. Go to **Project** > **Settings** > **Compiler** and select the Image craft compiler option.
4. Select **Build** > **Generate Configuration Files** for *'WirelessUSB-NL_Remote_Control'* Project.
5. Verify the Global Resource settings for WirelessUSB Remote Control project, as Figure 23 shows.

Figure 23. Global Resource for WirelessUSB Remote Control Device



6. Verify the GPIO pin-out settings for WirelessUSB Remote Control, as Figure 24 shows.

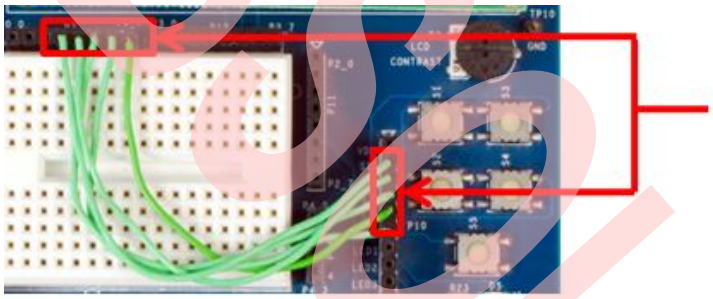
Figure 24. Pinout settings for WirelessUSB Remote Control Device

7. Build the project. Select **Build** > **Build** *'WirelessUSB-NL_Remote_Control'* Project or press **F7**.
8. For enCoRe V module, turn the SW1 switch downward on the CY3668 board; for the enCoRe II module, turn the SW1 upward.
9. Program the CY3668 DVK board with hex file from *"\ WirelessUSB-NL_Remote_Control\ WirelessUSB-NL_Remote_Control\ output\".*

4.4 CY3668 DVK Board Setup for WirelessUSB Remote Control

1. Attach the WirelessUSB-NL Radio module to P2, as Figure 19 shows.
2. Configure GPIO push buttons for remote control:
   - Wire up button S5 as Bind button, connect pin P0_6 on connector P7 to S5 on connector P10 with wire on CY3668 DVK board.
   - Wire up button S1 as media track play/pause button, connect pin P0_3 on connector P7 to S1 on connector P10 with wire on CY3668 DVK board.
   - Wire up button S2 as media track stop button, connect pin P0_4 on connector P7 to S2 on connector P10 with wire on CY3668 DVK board.
   - Wire up button S3 as volume increase button, connect pin P0_5 on connector P7 to S3 on connector P10 with wire on CY3668 DVK board.
   - Wire up button S4 as volume decrease button, connect pin P0_7 on connector P7 to S4 on connector P10 with wire on CY3668 DVK board.
3. A push button configuration is shown in Figure 25.

Figure 25. Push Button Configuration for WirelessUSB Remote Control Operation



4. Configure Radio, Power:
   - Place a two-pin jumper on J12.
   - Place a two-pin jumper on J6 connecting VRADIO and VREG.
   - Place a two-pin jumper on J2 and J3 connecting VDD and VREG.
   - J5 is used to select the fixed 3.3-V VREG or adjustable VREG from 3.66 V to 1.63 V. In this example, use fixed 3.3-V VREG.

   Jumper settings are shown in Figure 22.
5. Mount the LCD on P3 connector of CY3668 DVK board as shown in Figure 20.

4.5 Remote Control Functional Operation

1. Connect the CY3668 DVK configured as WirelessUSB Dongle to a PC using USB Mini B to USB A cable. WirelessUSB Dongle CY3668 DVK is configured as a bus-powered device.
2. WirelessUSB Dongle is enumerated as composite USB Device; verify that it enumerates an HID-compliant consumer control interface.
3. Connect the power adapter to a CY3668 board, which is configured as a remote control device.
4. Binding Process: Press the bind button S1 on the WirelessUSB Dongle and then press bind button S5 on the remote control to start the binding process.
5. LED2 on the dongle blinks after the bind process is initiated and keeps blinking until the bind process with remote control is complete; or, the timeout period (20 seconds) for the bind process expires.

   The remote control demonstration is ready for WirelessUSB Remote Control operation.
6. Open Windows Media Player and play any multimedia track.
7. Use Play/Pause, Stop, and Volume increment/decrement operation by pressing buttons S1, S2, S3, and S4, respectively.

Figure 26. Operational Setup of WirelessUSB Dongle Device

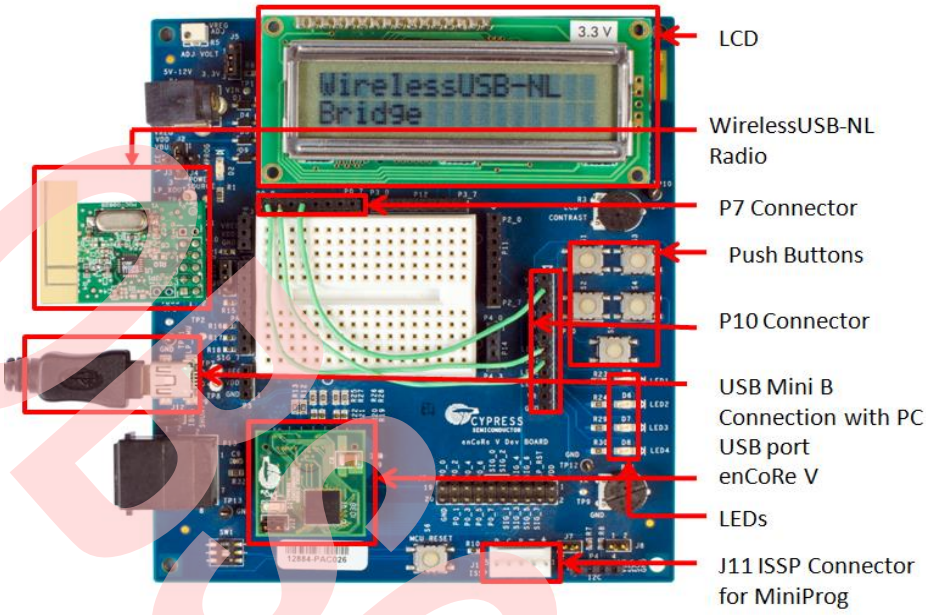
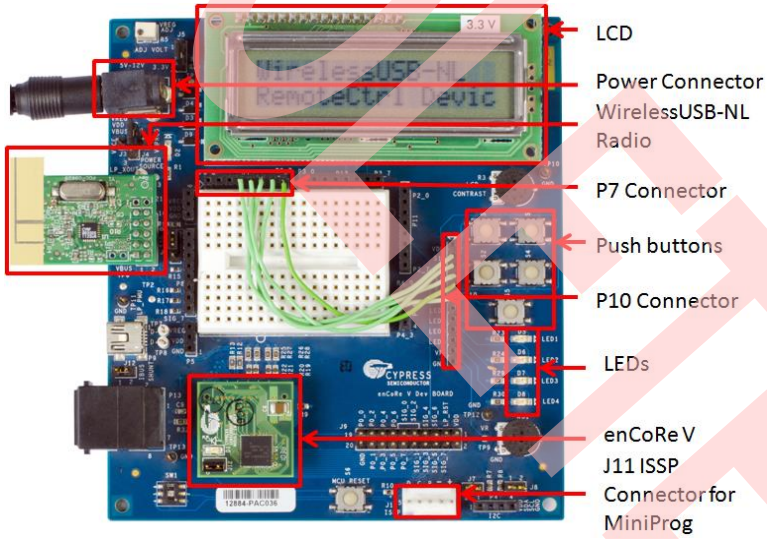


Figure 27. Operational Setup for WirelessUSB Remote Control Device



5 Summary

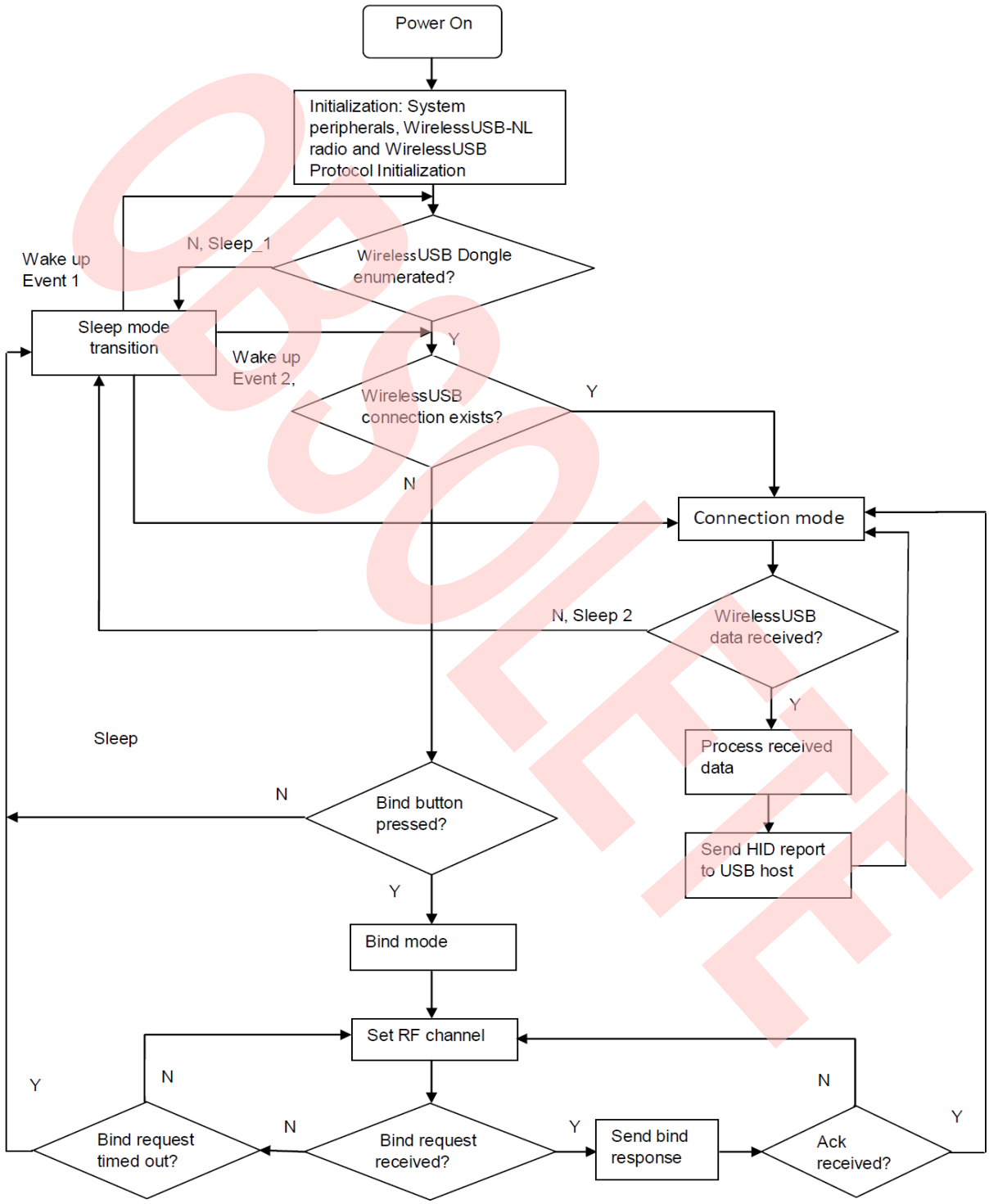
The Wireless Remote Control HID device uses a 2.4-GHz low-power radio transceiver to communicate wirelessly over the physical radio link with the WirelessUSB Dongle. The WirelessUSB RF-based Remote Control overcomes two major problems associated with IR-based remote control: line-of-sight and power consumption.

You can use the RF-based remote control for a home entertainment system by adding other consumer control interface functions.

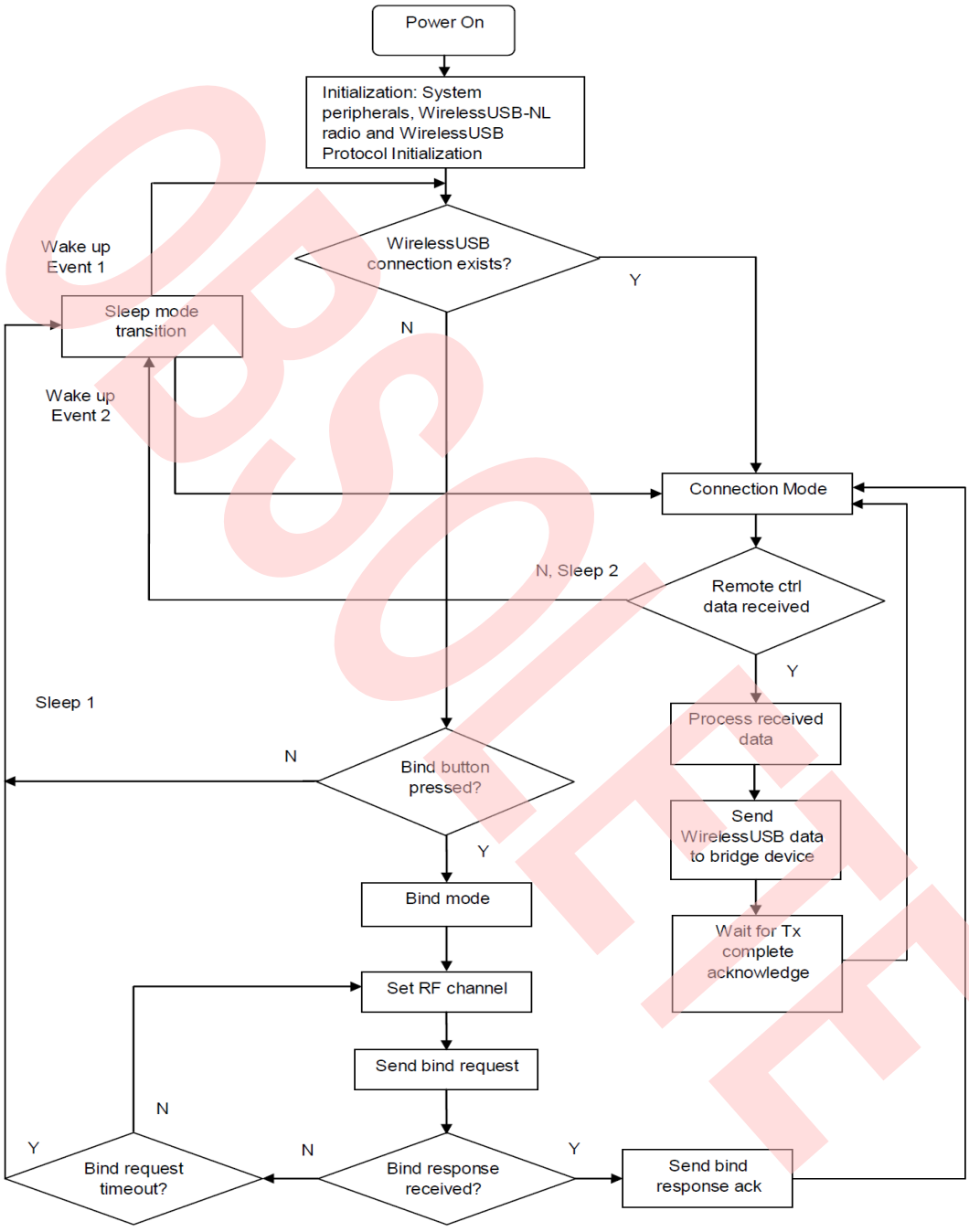
6 Related Documents

Document	Title
USB HID1.11	USB HID Specification
USB HUT1.12	USB HID Usage Table Specification
001-76173	CY3668 WirelessUSB™-NL Development Kit Guide
001-32519	enCoRe™ V CY7C643xx, enCoRe™ V LV CY7C604xx TRM, Document No. 001-32519
001-72412	WirelessUSB™-NL Radio User Module Datasheet (PSoC Designer)
001-13629	PSoC USB Datasheet
001-70689	WirelessUSB™-NL Radio Driver API Document
001-70688	Enhanced AgileHID™ Protocol API Document

7 Appendix A. WirelessUSB Dongle Application Flow Chart

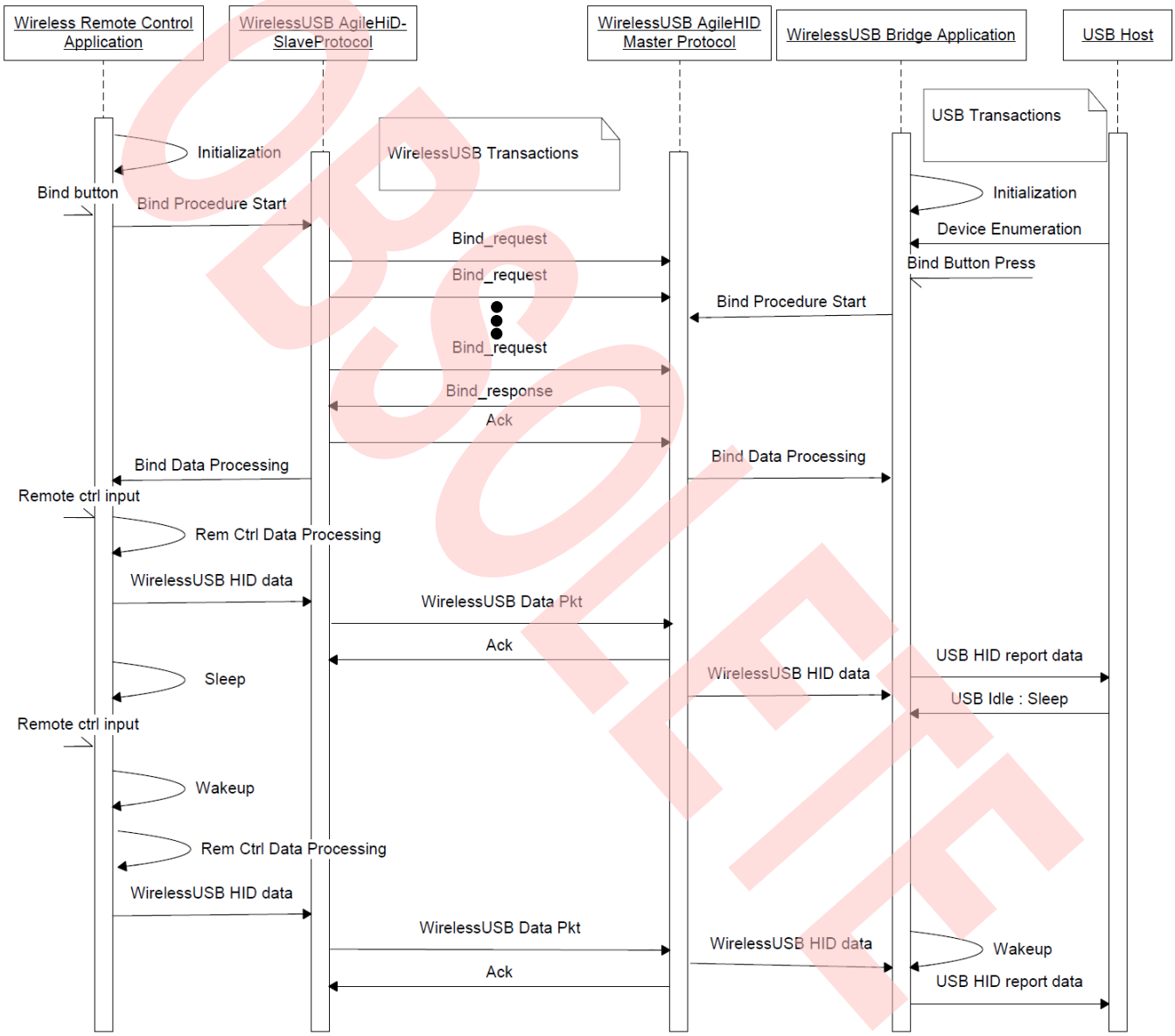


8 Appendix B. WirelessUSB Remote Control Flow Chart



9 Appendix C. Message Sequence Chart for WirelessUSB Remote Control Transactions

This chart captures the WirelessUSB HID device binding procedure and the WirelessUSB Data transactions with the WirelessUSB Dongle



Document History

Document Title: AN83281 - Developing RF-Based Remote Control Using WirelessUSB™-NL

Document Number: 001-83281

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3846886	DKSH	12/19/2012	New Spec.
*A	4779836	ANKC	07/08/2015	Updated template
*B	6234739	ANKC	07/09/2018	Obsoleting the spec

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com