Abstract

This Application Note describes the integration of the Cypress WirelessUSB™ and PSoC® CapSense interface technologies into a wireless remote control with capacitive sensors. This implementation demonstrates a WirelessUSB two-way simple protocol, a mixed-signal PSoC device, and implementation of the capacitive switch design using the CSR user module in PSoC Designer™. The proposed device can be used as a design starting point for the many popular wireless devices that markets demand today.

Introduction

A wireless remote control has the following advantages over traditional infrared signal (IR)-based remote controls:

- It does not require face-to-face device directivity.
- It allows for control through a wall.
- It can operate with many devices simultaneously.
- It allows for the use of several wireless devices in a shared room space.

Capacitive sensors have advantages over traditional mechanical switches, including greater reliability and lower cost. Integration of the Cypress WirelessUSB and PSoC CapSense technologies into a wireless remote control with capacitive sensors combines all these advantages in a single device. The unique architecture of the PSoC allows this device to be built at a very affordable price with external components kept to a minimum. The wireless remote control with capacitive sensors can be used as a remote control for various devices (TV sets, air-conditioners and similar home appliances), in distance-measuring equipment, medical systems and related devices.

A block diagram for this wireless remote control is shown in Figure 1. Technical specifications are listed in Table 1.
The proposed wireless remote control device with capacitive sensors is described in the following sections:

Device Schematic

WirelessUSB Two-Way Protocol

Appendix

Device Schematic

A complete schematic of the remote control device is shown in Figure 4 (Appendix).

Using PSoC Designer's CSR User Module, the PSoC device U1 detects the presence of a finger through glass, plastic, acrylic or many other non-metallic materials on a simple PCB trace through a CapSense button. The PSoC SPI User Module permits communication by means of the WirelessUSB radio transceiver, CYWUSB6934. The firmware modules for working with this radio transceiver were taken from the WirelessUSB LS development kit, CY4632 "WirelessUSB LS™ Keyboard-Mouse Reference Design Kit with KISSBind™." The PSoC device also implements a two-way communication protocol and other task-oriented control functions. For more information about the PSoC application, see the associated project files.

The WirelessUSB radio transceiver, CYWUSB6934, must be connected to the connector J1. The connector J2 provides the in-circuit PSoC programming firmware.

The remote device user interface (panel) is accessed through an array of CapSense buttons (BRight, BDown, BOK, BUp, BLeft), LEDs (D1-D5) and buzzer (Y1). When the user touches the buttons on the remote device, the corresponding LEDs light up on both the remote and receiver devices (if the devices are connected). This provides for remote control of wireless LEDs by capacitive sensors. The buzzer Y1 allows one to search the remote device by toggling the BFind switch on the receiver. In this mode, the buzzer on the remote device beeps and all the LEDs light up. The proposed implementation is a prototype, but can be used as a design starting point for wireless devices required by today's market. It is also possible to add capacitive sliders, rotating wheels, and other options for a more feature-rich user interface.

To provide a processor power supply from a low voltage level, the boost converter U2 is used. For example, the low-cost XCY672S Torex series of DC/DC converters with output voltage equal to 3.06V can be used. The Torex DC/DC converters are optimized to extend the battery life for wireless peripherals that are designed with the Cypress WirelessUSB radio SoC and microcontrollers (see http://www.torex.co.jp/english/).

This implementation can easily be adapted for the CYWUSB6953 WirelessUSB PROc (Programmable Radio System-on-Chip). For more information about the WirelessUSB Flash Programmable MCU + Radio see "Wireless Products" on the Cypress web site.

The schematic of the receiver is shown in Figure 5 (Appendix). The schematic is similar to the schematic for the remote control device in Figure 4, described earlier in this Application Note.

WirelessUSB Two-Way Protocol

A WirelessUSB two-way communication protocol is shown in Figure 2. For more information about WirelessUSB communication protocols, see CY4632 "WirelessUSB LS™ Keyboard-Mouse Reference Design Kit with KISSBind™" on the Cypress web site. Note that existing two-way WirelessUSB communication protocols are designed only for stationary receivers. The receiver in this application is stationary and its power consumption is not minimized. Only the remote control is designed to be a portable device with low power consumption.

By adding time synchronization between the remote control and the receiver—which allows them to remain in sleep mode most of the time—both devices are portable. A user can easily add this time-synchronization feature.

Wireless communication between devices requires the use of the same frequency, PN code, and code seed. Therefore, at the start of communication, an automatic channel-selection procedure is performed. The remote device selects the first available channel, sets the default code seed and PN code, repeats transmission of the connect-request data packets, and listens for the connect-response packets. If the response time-out has expired and a response is not received, then the remote device goes to the Start state, selects the next available channel, and again attempts a connection.

Table 1. Specifications for Wireless Remote Control with Capacitive Sensing

<table>
<thead>
<tr>
<th>Item</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supply Voltage (2 AA, AAA Batteries)</td>
<td>0.9…3.2V</td>
</tr>
<tr>
<td>Remote Device Power Consumption (Average Values)</td>
<td></td>
</tr>
<tr>
<td>- Reconnect Mode</td>
<td>3 mA</td>
</tr>
<tr>
<td>- Data Transfer (Connected) Mode1</td>
<td>150-200 µA</td>
</tr>
<tr>
<td>Receiver Device Power Consumption (Continuous operation)</td>
<td>60 mA</td>
</tr>
<tr>
<td>Wireless Protocol</td>
<td>Two-Way WirelessUSB</td>
</tr>
<tr>
<td>Data Rate</td>
<td>64 kbps</td>
</tr>
<tr>
<td>Operation Range</td>
<td>10 meters or more</td>
</tr>
</tbody>
</table>

Note

1. The proposed implementation in this Application Note is not optimized for minimal power consumption. It is possible to significantly decrease current consumption by increasing the time interval between transferred packets.
The remote device also sets the default code seed and PN code. With these values, the remote control successively listens to all channels for the connect-request packet. When the remote device receives the connect-request packet, it immediately sends a response.

![Figure 2. WirelessUSB 2-Way Communication Protocol](image)

After the remote device sends the connect-response packet and the receiving device receives this packet, the two devices set the new selected code seed value TX_SEED and try to ping the channel for verification. This random unique code seed value protects the data from being interpreted by similar devices as their own. After a correct pinging response, the devices go into the Data Transmission state. In every state, both devices control the operation of the Time-out Expired event. When the Time-out Expired event appears, the devices go to the Start state and begin all over again. This is the main rule of the communication protocols.

To simplify the protocol, the PN code in this application is constant.

Remote device power consumption is minimized by using sleep mode between data transfers. Increasing sleep time (or time between packet transfers) decreases power consumption. However, in sleep mode the capacitive sensor interface is not active. This decreases the responsiveness of the device. An adaptive sleep mode addresses this problem. Every 125 milli-seconds (ms) the microcontroller wakes up and checks the button state. The device only wakes the radio module and sends a data packet if the button state has changed. Otherwise, the high-power-consuming radio module stays in sleep mode and the PSoC also returns to sleep mode.

To verify connection between devices and provide for data transfer from the receiver to the remote device, a data packet is sent at least every second. This is how the Find Remote Device signal is sent from the receiver.

### Conclusion

This Application Note describes the Cypress WirelessUSB and PSoC CapSense interface technologies. A wireless remote control with capacitive sensors is proposed. A PSoC project and implementation of the capacitive switch design using the CSR User Module in PSoC Designer have been developed. A simple two-way WirelessUSB protocol has been designed. The proposed implementation can be used as a design starting point for a wide variety of popular wireless applications.
Appendix: Schematics

Figure 3. Wireless Remote Control Device with Capacitive Sensors (Actual Size)

Figure 4. Schematic of the Remote Device
Figure 5. Schematic of the Receiver Device (Powered from Battery)
About the Author

Name: Oleksandr Karpin
Title: Post-Graduate Student
Background: Oleksandr received a computer-engineering degree in 2001 from National University "Lvivska Polytechnika" (Ukraine), and continues his study there as a post-graduate student. His interests include embedded systems design and new technologies.
Contact: karpinoo@ukr.net

In March of 2007, Cypress recataloged all of its Application Notes using a new documentation number and revision code. This new documentation number and revision code (001-xxxxx, beginning with rev. **), located in the footer of the document, will be used in all subsequent revisions. WirelessUSB is a trademark and PSoC is a registered trademark of Cypress Semiconductor Corp. "Programmable system-on-Chip," PSoC Designer, and PSoC Express are trademarks of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

© Cypress Semiconductor Corporation, 2006-2007. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.