A brief description of Coriandolo Radio

Craig Goldman, CoAutomation Inc.

Design Principles

Coriandolo Radio (CR) is a simple link designed to transfer data between remote sensing devices and a collecting device. The protocol is based upon four important principles.

1) Remote sensor devices have limited power resources.
2) Collection devices have significantly larger power resources.
3) The sensor devices must transfer data quickly to collecting devices.
4) The application allows design control of both the sensor devices and the collection devices. This means that the radio protocol need not be compatible with existing devices.

The protocol is intentionally minimal; this allows the protocol and support code to be very small. Wherever possible, we have pushed many concepts for data handling to optional "wrappers". This keeps the protocol from the burden of extra features that could slow down data transfer or use extra power. It also allows application programmers to customize the protocol for their needs.

Protocol Messages

CR imposes only the simplest possible composition for messages. All messages have three components – a **message length** byte, a 3-byte **Device Identifier** and a **data** field of 0 to 250 bytes. The messages themselves are the payload component of a radio packet supported by hardware. The Device Identifiers need not be unique among all CR devices; Device Identifiers only need to be unique for the situation since the application runs both sides of the data transfer.

To allow for rapid transfer of data, the CR code provides a bit of management for messages ready to transmit and empty buffers ready for reception. The code also maintains a field for each message so it can determine which message is the oldest; this allows transmitting the oldest message first. The same field is used to provide the application with received messages in the order of reception.

Protocol Operation

There are two types of devices – **Sensor** devices and **Base** devices. The Sensor device controls most aspects of the transfer of data so it can minimize power and maximize transmission rates. The Base collects data from one or more Sensors.

CR is a "frequency hopping" protocol operating in the 2.4GHz radio band. It uses five different radio frequencies. At any time, the Sensor and Base devices may be transmitting or receiving on any one of the five frequencies. Over time, all five frequencies will be equally used.

Coriandolo Radio data transfer has two distinct phases – **Announcement** and **Exchange**. The Sensor device begins the protocol by transmitting an Announcement; the Announcement message contains the Device Identifier of the Sensor. After transmitting the message, the Sensor listens for a response. If a message is received, the Announcement phase ends and the protocol changes to the Exchange. If no message is received before a timeout period ends, the Sensor "hops" to a new radio frequency and transmits again. An Announcement may be transmitted on all five frequencies if a Base device does not respond. The Base device, having access to larger capacity power sources listens most of the time; it sets the Rx frequency, activates the receiver and waits for a Sensor device to transmit an

Announcement. If no message is received before the end of a longer timeout period, the frequency is changed and the Base begins "listening" again.

The Announcement may be a specific message or it can be a minimal **Default Announcement**. The Default Announcement is a four-byte message that contains just the length byte and the 3-byte Device Identifier of the Sensor. The Default Announcement is preferable is most situations; typically no Base device is in-range when the Sensor makes the Announcement.

A Base device, having correctly received a message from a Sensor, always responds to that Sensor. The response message takes one of two forms. If the Base device has a transmit message with a Device Identifier that corresponds to the one just received, it will transmit the oldest such message back to the sensor. If the Base device does not have a matching message to transmit, it will transmit a **Default Acknowledge** message which contains just four bytes. The Device Identifier of the response message from the Base is the Device Identifier of the Sensor. This response ends the Announcement phase. It should be noted that the Device Identifier of the Base device is not transmitted as part of the CR protocol.

After the Sensor transmits an Announcement and receives a message from the Base, the Sensor can optionally begin an **Exchange** of messages. The Sensor transmits the oldest message, then listens for a response from the Base. The Exchange continues as long as the Sensor has messages to send and has received a response. Eventually, all available messages have been transmitted, the Sensor runs out of empty receive buffers or a radio response is not received.

Low-power, High transfer rate

The protocol employs several features to minimize power usage by the Sensor and increase data transfer rates from the Sensor to the Base. First, the Sensor Announcement is typically four bytes. This minimizes Sensor transmit power when the Base is not present. Second, the protocol allows the application to determine how often the Sensor "Announces." Currently, we favor transmitting an Announcement every four seconds. We have implemented applications that have rapid Announcements when the Sensor detects a Base device and has lots of data to transfer. The protocol also allows Sensors to announce less frequently, for example at night, when "drive-by" transfers are not likely. Third, data throughput from Sensor to Base is improved by using only the short Default Acknowledgement by the Base. Fourth, the protocol supports transfer of up to 250 bytes of data per message; this is much larger than some other low-power radio protocols (e.g. BLE).

Implementation

CR has been implemented on a Nordic Semiconductor nRF51 device. The protocol occupies less than 8 Kbytes of flash and requires less than 4 Kbytes of RAM for statics and stack (including the empty structure to support buffer management).