1.3 SS100 Operational Description

Spread spectrum systems come mainly in two varieties: Direct sequence and frequency hopping. The SS100 is the latter. Frequency hopping spread spectrum systems are similar to a standard narrow-band fixed-frequency system in the way that communications take place. The main difference is that frequency hoppers are constantly changing RF channels within a given range of the RF spectrum. So how do they stay synchronized?

First, a little background in some of the basic regulations for 902-928 MHz FHSS systems operated in North America. 1) The order in which the device changes channels must be pseudo-random. 2) For a 900 MHz ISM device to operate at the maximum 1 watt RF power level, it must use at least 50 channels in its hopping pattern. The SS100 has 256 such hopping patterns consisting of 50 channels each. 3) Each channel must be used equally to comply with the rules for non-stop communications. The SS100 changes RF channels approximately every 20 ms with default settings. This means it takes about 1 second for it to use all 50 channels in a given hopping pattern.

There are different ways to achieve synchronization of the radios. The method used by the SS100 is referred to as master/slave communications. A master radio is constantly sending out synchronization packets at the beginning of each channel change. If a slave radio is not yet synchronized with the master, it will start changing RF channels at a much slower rate than that of the master. It could completely stop changing channels and wait for the master to catch up to it, but changing channels occasionally helps prevent non-synchronization in cases where there is interference on some of the channels. In this manner, with good communications, synchronization should occur within a maximum of 1 second or the duration of the entire hopping pattern. Once the slave radio receives a synchronization packet and verifies that it came from its configured master, it then starts to change channels in tandem with the master. If the slave doesn't hear from the master on a particular channel, it starts counting. If this happens too many times in a row, it assumes it has lost synchronization and starts the process all over again.

One of the unique aspects of the SS100 system is its speed and reliability of data collection from the remote systems. The basic data collection algorithm is described as follows:

- Upon waking from a configured sleep time, the master builds a table with the number of configured retries for each of the remotes in a given network.
- The base embeds the ID of each slave in the network in the synchronization packet sent to each slave, starting with ID = 0 and ending with the highest configured remote ID in the network, at which time the loop starts back at ID = 0.
- After a synchronization packet is sent, the master switches to receive mode waiting for a response from the slave with the ID contained in the synchronization packet. Once the slave receives a synchronization packet with its configured embedded ID, it switches to transmit mode, reads its configured sensors, and transmits a 'sensor' packet to the master.
- By the time the receiver actually responds with the transmission, a channel change has taken place. This is why the master only sends synchronization packets every other channel change. If the master successfully receives the 'sensor' packet from the slave, it is transmitted out of the master’s serial port to whatever device it is connected to (a PC or some other device). The master also records that this slave has responded. If the master does not receive a response from the remote slave, it continues to the next ID.
- If the current slave ID has already been called successfully for this collection cycle, the master sends this slave a 'sleep' packet containing the amount of time to the next collection cycle. Upon receiving the 'sleep' packet, the remote slave then turns its radio circuitry off for the specified time.
- This loop continues until the master has either successfully called all slaves or exhausted all retries for all remote slaves, at which time the master will shut down its radio circuitry for the remainder of the configured sleep time.

This algorithm allows for data collection at a rate of up to 25 stations per second, efficient battery usage, distributed retries, and small packet size / short hop interval for increased reliability.