Introduction

The purpose of this guide is to aid in the software configuration of the Smart Radio. The Smart Radio is available in three different form factors: Embedded, External, and Wearable. As each form factor is designed to cater to a particular use case, there are differences in the hardware interfaces between the three designs. Go to https://doodlelabs.com/products/smart-radio/ for more information. This guide is organized as follows.

1. Smart Radio Settings
2. System Configuration using UCI
3. Smart Radio Ports
4. Smart Radio Network Configurations
5. Diagnostics Tools
6. Application Notes
7. Troubleshooting

Smart Radio Settings

Default Network Configuration

The Smart Radio runs Doodle Labs Mesh Rider® OS, a customized version of OpenWrt with enhancements. These enhancements are useful for applications requiring low-latency command-and-control transmission, HD Video, high-throughput and long range. Fig 1 shows the default network configuration of Smart Radio.
Fig. 1 Default Network Configuration

Not all interfaces are available in every model. Table 1 shows the available interfaces for the different radio form factors. ETH1 is configured by default as a configuration interface which is not remotely accessible, but it can be re-configured if necessary.

Table 1: Available Interfaces for Smart Radios

<table>
<thead>
<tr>
<th>Interface</th>
<th>Embedded</th>
<th>External</th>
<th>Wearable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh Rider Radio</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Ethernet (ETH0)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Ethernet (ETH1)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>USB Device Port</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>USB Host Port</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>WiFi Radio</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Mesh Rider Interface

The Mesh Rider interface is the main wireless interface and is available on all Smart Radio products. Because it uses a proprietary wireless protocol, it is only accessible from other Mesh Rider devices.
**Configuration Guide**

**ETH0 Interface**
ETH0 is a standard Ethernet interface and is bridged to the Mesh Rider Interface.

**ETH1 Interface**
The ETH1 interface is a standard Ethernet interface not bridged to the Mesh Rider interface. It is normally used for device configuration. It is possible to modify the interface so that it is bridged to ETH0 if desired.

**USB Device Port**
The USB Device port is an Ethernet over USB interface and can be connected to USB host ports like those found on PCs, or USB OTG ports like those on Smart Devices. It is bridged to the Mesh Rider interface.

**USB Host Port**
The USB Host port is set up as an Ethernet over USB interface and can be connected to USB device ports or USB OTG ports like those found on Smart Devices. It is bridged to the Mesh Rider interface. It’s functionality can be extended by installing additional USB drivers on the Smart Radio, like USB HID drivers.

**WiFi Radio**
The WiFi radio is bridged to the Mesh Rider Interface and provides standard WiFi connectivity to devices such as PCs and Smart Devices. The default SSID is DoodleLabsWiFi and the password is DoodleSmartRadio.

**Configuration Access**
For a full list of configuration access protocols supported by the Smart Radio, please visit the Doodle Labs Technical Library ([https://doodlelabs.com/technologies/technical-library/](https://doodlelabs.com/technologies/technical-library/)). In this guide, we will use the SSH protocol which uses network port 22, and the web GUI which uses network port 443. The web GUI is accessible from standard browsers like Google Chrome and Mozilla Firefox. An SSH client is built into most desktop Linux distributions, as well as Windows 10. You can use SSH for terminal access to the Smart Radio with the command,

```bash
$ ssh root@<IP ADDRESS>
```

Or you can remotely execute a command in the Smart Radio with

```bash
$ ssh root@<IP ADDRESS> `<COMMAND>'
```

The next section describes the IP configuration.

**IP Addressing**
There are three IP addresses defined by default.
1. A static IP address in the 10.223.0.0/16 subnet. This IP address differs from device to device and can be used for remote configuration or sending data to or from the Smart Radio. The IP address can be calculated from the MAC address. If the IP address is 10.223.x.y and the MAC address is 00:30:1A:4E:A4:3E, the x is equal to the decimal value of 0xA4, or 164, and y is equal to the decimal value of 0x3E, or 62. The full IP address is therefore 10.223.164.62/16. On newer devices, this IP address is printed on the label.

2. A DHCP client. This is a dynamic IP address which is automatically assigned when there is a DHCP server on the network.

3. A static IP address at 192.168.153.1/24. This is a fixed IP address which is the same for all radios. This IP address is designed to be used during initial configuration only and should never be used when more than one radio is on the network.

If you are unsure of the IP address on the 10.223.0.0/16 subnet, then the best option is:

1. First change the IP address of your host system to 192.168.153.10/24 (any valid IP address in the 192.168.153.0/24 subnet not equal to 192.168.153.1 will work). The method to do so varies depending on the OS and whether you are using the WiFi interface, or the USB device interface. If you are connecting over WiFi for example, then you probably need to have your devices associated with SmartRadioWiFi before changing the IP address.

2. Next run a command on the Smart Radio over SSH to print out the IP address on the 10.223.0.0/16 subnet.

```
$ ssh -q -o stricthostkeychecking=no root@192.168.153.1 'ip addr | grep 10.223'
```

```
itnet 10.223.170.187/16 brd 10.223.255.255 scope global br-wan
```

Once you have the 10.223.0.0/16 IP address, you can change your host system’s network configuration to use a static IP address on this subnet.

**Gateway Modes**

There are two gateway modes defined in the simpleconfig menu, WDS AP with gateway, and Mesh with gateway. In these modes, a DHCP server runs on the Mesh Rider interface at 10.222.1.1/16, and the 192.168.153.1/24 fallback IP address is disabled.

**Wireless Settings**

**Standard Radios**

The radio’s Mesh Rider wireless settings can be found by navigating to network → wireless in the GUI. There should only be one radio interface present if you setup your network using the
Configuration Guide

Simple Configuration menu (discussed later). To modify the wireless settings, click Edit. You should see a page similar to the one below. We recommend the following updates.

- For point-to-point networks such as a Control Station controlling a single UAV, or Robot, **Enable** Dynamically Adjust txpower based on neighbor sounding (Transmit Power Control).
- At power up, the Smart Radio will scan the environment and choose the best channel for the environment. A different channel can be chosen manually.
- For 2.4-GHz ISM-band radios, use a channel bandwidth of 15-MHz to avoid normal WiFi interference. This is especially important in urban areas.
- Change the Mesh ID and under Wireless Security
- Change the password.

![Fig. 2 Wireless Settings](image)

Network Interface Settings

Network interface settings can be modified in the **Network → Interfaces** tab. Below are some common changes you may want to make.

- Enable a DHCP server on one of the radios.
  - Click Edit next to the **WAN** interface. You should see a page similar to the picture below. Change the **Protocol from DHCP to Static Address**. Add an IPv4 address and netmask of your choosing, then scroll to the bottom of the page, and un-check **Disable DHCP for this Interface**. Click **Save & Apply**.
• You may also wish to enable a DHCP server on the LAN interface (ETH1). The steps are similar to those above.
• Change the default static IP address of ETH0. To do this, edit the WAN2 interface, and change the static IP address to your liking.

![Fig. 3 Network Interface Settings](image)

Mesh Settings

The default radio network configuration is a mesh, and the configuration settings can be accessed at Network → Mesh Configuration. However, this menu is only accessible in the Advanced Settings which is opened by clicking the Advanced Settings button in the bottom of the left hand side menu bar. Fig. 4 shows the Mesh Configuration Page. In general, you should not need to change most of the settings except for:

• OGM Broadcasting Interval – This number affects the responsiveness of the mesh to routing changes. A smaller number is faster, but results in more network overhead. For large meshes, it is recommended to increase this.
• Bridge Loop Avoidance – This is only necessary if there are potential network loops created in the Ethernet backbone. For example, if two of the wireless mesh nodes are also connected over the wired backbone, a loop is formed, and Bridge Loop Avoidance should be enabled.
Group-Aware Multicast to Unicast Conversion – This option is necessary if your application uses the IGMP/MLD protocol for multicasting. It eliminates redundant data distribution and converts the multicast transmission to unicast.

![Mesh Configuration Page](image)

**Fig. 4 Mesh Configuration Page**

**Firewall Settings**

The Firewall configuration is located under Network → Firewall and can be modified over the GUI and CLI at /etc/config/firewall.

We shall use the iperf3 application as an example to demonstrate setting a firewall rule to allow access to port 5201 of the router. Navigate to the Firewall page, and click the Traffic Rules tab. In the “Open ports on router” section, enter the name, protocol, and port number as “Allow iperf”, “TCP”, and “5201” respectively so that clients is able to connect to port 5201.
Before port 5201 is opened, run `iperf` in server mode inside the Smart Radio.

```
root@smartradio-301a4e8646:~# iperf -s
```

If you try and connect to the `iperf` server from your local machine, you will get an error, `connect failed: Connection refused`. After opening the firewall at port 5201, you should be able to connect to the `iperf3` server.

Extensive information regarding Firewall configuration is available at the openwrt.org website.

**Traffic Prioritization**

Different types of traffic can be prioritized in the Traffic Prioritization menu. This is useful when operating in a crowded wireless medium. There are four different queues – Voice, Video, Best Effort, and Background. The Voice queue optimizes latency and may also be used for command and control, the Video queue optimizes throughput, the Best Effort Queue is essentially unoptimized, and the Background queue is for low-priority data.

In order to use these QoS features, open up the web GUI and navigate to `network → differentiated services`. The Smart Radio includes software to map different network protocols or ports to the various QoS queues. To do so, click `Enable Differentiated Services`, and add a classification rule to suit the application’s needs. For example, you can send all UDP traffic to the Video queue which is beneficial for video transmission.
URLLC (Ultra Reliable Low Latency Channel)
The Smart Radio includes protocol optimizations for URLLC applications as well as video optimizations. URLLC applications typically include command and control (C&C) data, but can be extended to any application requiring a reliable low latency. Assuming that we have a C&C application which uses network port 7000 over UDP. In the screenshot below, first click Optimize Command & Control for URLLC. Next click Add, and then change the new classification rule to use Port 7000, and set the DSCP value to CS6. The comment section can be filled if desired. Finally click Save & Apply and wait for the page to refresh.

![Fig. 6 Traffic Prioritization Settings](image)

**Advanced Settings**

By default, the web GUI only presents a small subset of all of the possible configuration options of the Smart Radio. For advanced users, you may wish to reveal all configuration menus by clicking the Advanced Settings button at the bottom left hand corner of the page.

**System Configuration using UCI**

**UCI Overview**

Openwrt uses the UCI system for device configuration. Most of the configuration files can be found here,
To show the current wireless configuration, type

```
root@smartradio:~# uci show wireless
wireless.radio0=device
wireless.radio0.type='mac80211'
wireless.radio0.path='platform/qca953x_wmac'
wireless.radio0.htmode='HT20'
```

Only the first four results are shown above. To change a setting, e.g. the wireless channel, use `uci set`:

```
root@smartradio:~# uci set wireless.radio0.channel='10'
```

This change will be held in memory but will have an effect when the network settings are reloaded using

```
root@smartradio:~# /etc/init.d/network reload
```

Changes to the configuration can be stored using

```
root@smartradio:~# uci commit
```

**Configuration Parameters**

Table 2 shows information on commonly used and the Doodle Labs Smart Radio specific configuration parameters. For full technical discussion of all of the different configuration parameters of the Openwrt system, the user is encouraged to explore the [OpenWrt documentation](https://www.openwrt.org).

**Table 2: Commonly used and Doodle Labs specific Configuration Parameters**

<table>
<thead>
<tr>
<th>Wireless</th>
<th>wifi-device section</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>txpower</strong></td>
<td>Specifies the transmit power in dBm. Note that transmit power is also limited by local regulations, and a rate-dependent target power.</td>
</tr>
</tbody>
</table>
## Configuration Guide

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>channel</strong></td>
<td>This is the wireless operating channel.</td>
</tr>
<tr>
<td><strong>fes_disabled</strong></td>
<td>This will put the front-end-system in sleep mode (low power)</td>
</tr>
<tr>
<td><strong>chanbw</strong></td>
<td>Sets the channel bandwidth in MHz. Only model specific valid bandwidths can be used. Supported bandwidths are 3/5/10/20/26 with 0 being 20 MHz.</td>
</tr>
<tr>
<td><strong>distance</strong></td>
<td>This should be set to a value higher than the maximum expected transmission distance in meters.</td>
</tr>
<tr>
<td><strong>rxantenna</strong></td>
<td>Used to selectively enable the two antennas in a comma separated format. For example, ‘2’ will enable only the second antenna and ‘1 2’ will enable both antennas for reception.</td>
</tr>
<tr>
<td><strong>txantenna</strong></td>
<td>Used to selectively enable the two antennas in a comma separated format. For example, ‘1’ will enable only the first antenna and ‘1 2’ will enable both antennas for transmission. DO NOT DISABLE THE FIRST ANTENNA.</td>
</tr>
<tr>
<td><strong>nf_override</strong></td>
<td>Sets a noise floor override in dBm, which can be useful to ignore interferences. Default value is approx. -95.</td>
</tr>
</tbody>
</table>

### wifi-iface section

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>mesh_fwding</strong></td>
<td>Set to ‘0’ to disable 802.11s forwarding. Mesh forwarding is typically handled by batman-adv.</td>
</tr>
<tr>
<td><strong>mesh_ttl</strong></td>
<td>Set to 1 to disable further forwarding in 802.11s. Forwarding typically handled by batman-adv</td>
</tr>
<tr>
<td><strong>mcast_rate</strong></td>
<td>Specify the multicast modulation rate in mesh networks, in bps (e.g. 12000 = 12 Mbit/s). If set higher, the range is decreased but the faster links are preferred.</td>
</tr>
</tbody>
</table>

### Diffserv

#### general section

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>enabled</strong></td>
<td>Set to 1 to enable diffserv</td>
</tr>
<tr>
<td><strong>optimized_cc</strong></td>
<td>Set to 1 to enable URLLC operation for the Command &amp; Control and Voice Queue</td>
</tr>
<tr>
<td><strong>optimized_vi</strong></td>
<td>Set to 1 to enable URLLC operation for the Video queue</td>
</tr>
</tbody>
</table>

### mark section

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>dscp</strong></td>
<td>Select the DSCP queue, e.g. BE = best effort</td>
</tr>
<tr>
<td><strong>proto</strong></td>
<td>Define the protocol, e.g. icmp, tcp or udp</td>
</tr>
<tr>
<td><strong>srchost</strong></td>
<td>The source and the destination host are IPv4 address (e.g. 192.168.1.2) can be used to apply rules only to a specific host. A network mask can be used to match entire networks. 192.168.1.0/24 will match traffic with an IP address in the range of 192.168.1.1 to 192.168.1.254. Blank fields will match any address.</td>
</tr>
<tr>
<td><strong>dsthost</strong></td>
<td></td>
</tr>
</tbody>
</table>
Configuration Guide

| **ports** | Affected ports for this rule. Valid ports in rules can be in the range 1-65535. Port ranges can be given in the form of a single port, an inclusive range <start-port>-<end-port> or as a comma separated list <port>,<port>. Both notations can not be mixed. |

| **Network** |  |
| **bat0.bridge_loop_avoidance** | Set to 1 if there is a chance that devices can be connected via Ethernet and Mesh at the same time |
| **bat0.orig_interval** | Interval of batman-adv originator messages (management frames). Default is 1000. Decrease to more frequently update the network topology, e.g. for high mobility nodes. This may limit the maximum # of nodes on the network. |
| **mesh_dev.num_bcasts_own** | Number of times a broadcast originating from the radio is transmitted (default 3) |
| **mesh_dev.num_bcasts_other** | Number of times a broadcast originating from another radio is re-transmitted (default 3) |

**Smart Radio Ports**

**UART**

The UART port is a standard TTL level 3-pin serial port (RX, TX, GND). The UART port can be used directly with flight controllers such as the PixHawk 4. It is also possible to setup a network to serial relay, or a direct serial to serial link over the network. Please refer to Doodle Labs’ Application Note on using the serial port for further details.

**USB Device Port**

The USB Device Port only supports Ethernet over USB protocols. It works with Laptops and Tablets as long as they support Ethernet over USB. The USB Device Port uses the ETH0 interface logically.

**USB Host Port**

The USB port is a host port and is pre-configured as an Ethernet over USB interface. When plugged into a USB device or OTG port with a compatible Ethernet over USB configuration, a new interface, USB0, will be instantiated and bridged to the WAN interface. It is then possible to access the Smart Radio over the web browser or SSH at the default static IP address defined earlier. It is possible to install other USB packages to include other USB functions for your application.
The Smart Radio has a USB hub inside and USB ports are accessible on the main and secondary connectors. USB0 on the main connector does not have a 5-V supply associated with it.

**GPIO**

In hardware version J and later, GPIOs are available for programming. The GPIOs are provided by an on-board MCU and interfaced to the Operating System over USB HID. A program “sr-ctrl-usb” is pre-installed in the system and is used to access the GPIOs. The general syntax for accessing the MCU is

```
root@smartradio:~# sr-ctrl-usb <r/w> <param #> <value>
```

`r/w` is either “0” for read or “1” for write. The parameter numbers for the three GPIOs are tabulated below. The possible values which can be read or written are “0” and “1” and correspond to 0-V and 3.3-V. Table 3 summarizes GPIO parameters.

**Table 3: GPIO Parameters**

<table>
<thead>
<tr>
<th>Param #</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Logic level of GPIO1, 0 is GND and 1 is 3.3-V</td>
</tr>
<tr>
<td>7</td>
<td>Logic level of GPIO2, 0 is GND and 1 is 3.3-V</td>
</tr>
<tr>
<td>8</td>
<td>Logic level of GPIO3, 0 is GND and 1 is 3.3-V</td>
</tr>
<tr>
<td>11</td>
<td>Direction of GPIO1, 0 is Output and 1 is Input</td>
</tr>
<tr>
<td>12</td>
<td>Direction of GPIO2, 0 is Output and 1 is Input</td>
</tr>
<tr>
<td>13</td>
<td>Direction of GPIO3, 0 is Output and 1 is Input</td>
</tr>
</tbody>
</table>

As an example, if we want to write a level 1 to GPIO2, we first set the direction of GPIO2 to 0 and then we set the logic level to 1.

```
root@smartradio:~# sr-ctrl-usb 1 12 0
root@smartradio:~# sr-ctrl-usb 1 7 1
```

We can read the logic level of GPIO3 with,

```
root@smartradio:~# sr-ctrl-usb 1 13 1
root@smartradio:~# sr-ctrl-usb 0 8
```
Note that the GPIO directions are reset to input when the MCU is reset. When the MCU is reset, a reset flag is raised at param # 13 and it should be reset to zero by the user.

**Wake-Up Timer**

For very low power monitoring applications, the Smart Radio can be configured to completely shut down and wake up on a timer. In order to conserve power, the CPU itself shuts down and it requires around 30 seconds to boot up again. The command to shutdown the system is,

```
root@smartradio:~# sr-ctrl-usb 1 2 <time in seconds>
```

where `<time in seconds>` is the amount of time that the CPU should shut down for. This feature is only available in -J hardware variants.

**Firmware Upgrade**

The Mesh Rider OS can be upgraded using either the Web GUI, or the Linux console. Connect the host PC to the ETH1 port of the Smart Radio.

**Console Method**

1. Copy firmware from PC to the module:

```
scp firmware-sysupgrade.bin root@192.168.1.1:/tmp/
```

2. Login to Smart Radio by ssh `root@192.168.1.1`

3. Use the `sysupgrade` command to update the firmware.

```
sysupgrade -v /tmp/firmware-sysupgrade.bin
```

4. Wait until the update is complete. DO NOT CYCLE THE POWER SUPPLY UNTIL THE FIRMWARE HAS BEEN UPDATED!

**Web GUI Method**

1. Enter the following into the address bar of your browser

   `https://192.168.1.1/cgi-bin/luci/admin/system/flashops`

2. At the bottom of the page, de-select the “keep settings” button

3. In the “image” field, click browse and select your firmware image

4. Click Flash Image

5. After the system has verified the image, click Proceed
Smart Radio Network Configurations

The Smart Radio supports many different network configurations. The Simple Config menu makes it easy to configure the radio. The Simple Config menu is accessed in the web GUI at network → simple config.

- Mesh (Factory Configuration)
- Mesh Gateway
- WDS Access Point
- WDS Client
- WDS Access Point Gateway
- Multi-Radio Mesh

The wearable form factor of the Smart Radio includes a standard WiFi radio which can be independently configured as either a WDS AP or WDS Client device.

Mesh Configuration

Fig. 7 shows the mesh configuration. This is the factory setting of the Smart Radio.

In this configuration, a Router is connected to one of the Smart Radios at ETH0, which acts as an Internet Gateway for the mesh. The Router runs a DHCP server which provides IP addresses to all of the host devices in the mesh. In the absence of the Gateway Router, you may use the Mesh Gateway configuration described next. If mesh network is not required, we suggest using the WDS AP/Client mode to get a slightly higher throughput.
There are two networks configured within each Smart Radio. The first is the mesh network, and the second is the configuration network.

- The mesh network consists of all Smart Radios and host platforms which join the mesh. ETH0 is bridged to the mesh network. The firewall is set up to allow input access using SSH or the web GUI. The mesh network has two IP addresses.
  - The first is a DHCP client.
  - The second is a static IP address equal to 10.223.x.y where x.y is the last four hexadecimal digits of the wireless MAC address. For example, a unit shipped with MAC address 00301A4E7AB0 has the last four HEX digits 7AB0 and x.y is equal to 122.176.

- The configuration network is meant to include only the Smart Radio and the host platform it is connected to via ETH1. The firewall is setup to allow all input access and it is designed to be a configuration port.
  - It has a static IP address equal to 192.168.1.1 and is not bridged to the mesh.

The Smart Radio can be accessed over the command line with

```bash
user@host-pc:~$ ssh root@<IP ADDRESS>
```

or using a web browser at the same IP address. <IP ADDRESS> depends on whether you are connected to ETH0 or ETH1. In Windows, use the PuTTY client. No password is set by default, and after setting one up, it is possible to setup public-key authentication following this guide.

**Mesh Gateway Configuration**

The Smart Radio also includes a gateway mode which obviates the need for an external router. The network diagram is shown in Fig. 8.
The Smart Radio in gateway mode (on the right) configures the network with a 10.223.0.0/16 IP address range and acts as a DHCP server. All other Smart Radios are in normal mesh mode. In order to access the gateway device, either access it wirelessly, or connect to ETH1 and use the 192.168.1.0/24 subnet.

**WDS AP/Client mode**

The WDS AP/Client network diagram is shown in Fig. 9. In WDS mode, traffic is bridged transparently between the router and the end device. As in the standard mesh mode, IP addresses are provided by the DHCP server running in the external router. The pre-defined static IP addresses of the Smart Radios are still active. WDS AP/Client mode establishes a star network topology. If mesh is not required, we recommend this mode since it is typically a little faster.

---

**Fig. 8 – Mesh Network with Gateway**
Fig. 9 – WDS AP/Client Network

WDS Gateway Mode
WDS Gateway mode is an extension of the WDS AP/Client mode where the WDS AP is also an internet gateway. The WDS AP Gateway runs a DHCP server and performs NAT between the local 10.222.0.0/16 subnet and the wider network.

Fig. 10 – WDS AP/Client Network with Gateway

Multi-Radio Mesh Mode
The Smart Radio supports direct connection between multiple radios using the ETH1 interface. In order to do this, change the configuration to multi-radio mesh mode in the Simple Config
menu. Note that in this mode, ETH1 is no longer the configuration port and cannot be accessed at 192.168.1.1. Two example network configurations are shown in Fig. 11.

**Fig. 11 – Multi-Radio Mesh Network**

In the above configuration, two radios are used to extend a link using two different frequencies to avoid having to share the medium. The two channels could be in the same band if desired; for example 2412 MHz and 2462 MHz.

**Fig. 12 Multi-Radio Mesh Network with Failsafe**

In the next configuration, the 915-MHz radios are used as a failsafe for the 2.4-GHz band. This is shown in Fig. 12. If the 2.4-GHz link cuts off due to it being out of range, or due to excessive interference, network traffic will switch to the 915-MHz link automatically. An important point in the diagram above is that traffic entering the mesh from one side always takes 2 hops before leaving the mesh on the other side. This balance allows the mesh to react significantly faster than if one route required more hops than the other.
Mesh + AP mode

In Mesh + AP mode, the same radio interface acts as both an AP and a Mesh node. As it is the same radio, physical settings such as channel bandwidth and channel must be the same. In Fig. 13, the two white Smart Radios are in Mesh + AP mode. The two robots act as WiFi clients and connect to the Mesh. The Mesh itself is formed by the black and the white Smart Radios allowing the two clients to have internet access. In order for this to work, the Mesh must operate at 2.4-GHz with a 20-MHz bandwidth. Note that the AP mode discussed in this section is standard WiFi AP mode rather than the WDS AP mode defined in the sections above.

![Fig. 13 Mesh + AP Mode](image)

Diagnostics Tools

Doodle Labs is working on introducing a set of radio diagnostics tools to aid in trouble-shooting. Currently the diagnostics tools are limited to the CLI, but GUI integration is planned in future. To enable and load the diagnostics tools, SSH into the radio, and run the following commands:

```
user@host-pc:~$ ssh root@<IP ADDRESS>
root@smartradio:~# /etc/init.d/sysutils enable
root@smartradio:~# /etc/init.d/sysutils start
```

The first tool which has been integrated is a pre-launch check. The user can specify a second radio to check a link against. The pre-launch check then checks the link quality of each individual antenna. An example call is

```
root@smartradio:~# sysutils checklink -d 10 -h 10.223.187.32 -g 6
```
This tells the pre-launch check to check the link to the radio at IP address 10.223.187.32. The distance between the radios is 10m, and the total antenna gain (TX + RX) is 3 dBi. The results are stored in a temporary file, /tmp/results.json, and an example output is

```json
{
    "distance": "10",
    "total_antenna_gain": "3",
    "frequency_band": "2442",
    "expected_RSSI": "-37",
    "RSSI1": "-41", "RSSI2": "-51",
    "throughput1": "21876687",
    "throughput2": "22352857"
}
```

Note that in this case, antenna 1 showed good RSSI relative to expectations while antenna 2 showed much poorer RSSI. This could simply be due to polarization mismatch, but it is worth investigating. Each antenna was able to obtain good throughput (~21 Mbps per antenna).

**Application Notes**

Doodle Labs has developed many application notes for Smart Radio. They are available in the [Technical Library](#) section of the website. Below list shows some of the commonly used application notes.

1. Optimizing the wireless link Throughput
2. Optimizing the wireless link Distance
3. Optimizing Latency for Command and Control Data
4. Video Streaming Tutorial
5. Expanding Smart Radio capabilities
6. Interference Mitigation Techniques
7. Remote Management of Smart Radio

**LED Blinking**

A simple LED blinking script is available in the Smart Radio which will cause one of the GPIOs to toggle ON/OFF. It starts by default, but you can disable it by logging into the radio over SSH and running

```
root@smartradio:~# /etc/init.d/startup_blink disable
```
For Wearable or Embedded (-J) radios, it is possible to choose which GPIO to blink (0, 1, or 2) and by default, none are selected. Edit the following lines of the file /usr/sbin/checkStartupBlink.sh to select the GPIO or change the blinking frequency.

```
# This variable could be init'd from config file
# Possible values are [0, 1, 2, 3]
gpio_selected=0

# This variable could be init'd from config file
# Blinking speed: 0.5sec, 1sec, 1.5sec, 2sec, 2.5sec, 3sec, etc.
blinking_hz=0.5
```

The GPIO will blink until the radio is associated with at least one station, after which it will hold steady.
Connect with Tech Support

Before connecting with Tech Support, please read the trouble-shooting information in our quick evaluation guides. The Diagnostics Tools described in the sections above may also be of use.

Otherwise, please feel free to contact us (https://www.doodlelabs.com/about-us/tech-support/tech-support-request-form/). Please describe your problem in as much detail as possible, and include photos of the setup if possible.

Configuration Backup

Additionally, please create a backup of your configuration and send it with your email. You can create a backup in the web GUI by navigating to system → backup/flash firmware and clicking Generate Archive.

If you are having connectivity issues between nodes, please fill out this table and include it in your email.

Table 4: Tech Support Form

<table>
<thead>
<tr>
<th>Question</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Name</td>
<td>RM-2450-2J-XM</td>
</tr>
<tr>
<td>Firmware version</td>
<td>firmware-2020-03, r4030-72047dd133</td>
</tr>
<tr>
<td>Test setup</td>
<td>Point-to-point, mesh, AP/Client</td>
</tr>
<tr>
<td>Type of Data</td>
<td>Video, Control signals, iperf, ping, multicast</td>
</tr>
<tr>
<td>Number of Antennas used</td>
<td>1 or 2</td>
</tr>
<tr>
<td>If one antenna, which one?</td>
<td>Antenna 0</td>
</tr>
<tr>
<td>Distance between nodes</td>
<td>100m</td>
</tr>
<tr>
<td>Height of the Antennas</td>
<td>3m</td>
</tr>
<tr>
<td>Number of nodes (mesh)</td>
<td>4</td>
</tr>
</tbody>
</table>