



# Mesa2 Android RFID Source Code Explanation

---

## Assumptions:

- You have already signed the RFID INDEMNIFICATION AGREEMENT
- You intend to write your own RFID application for the Android Mesa2

## Your responsibilities:

- Regulatory compliance – This is the reason you signed the agreement, to control transmit power levels, operate in the correct frequency (region), and possibly put cellular into airplane mode.
- Control power during sleep – Your application is responsible for putting the RFID expansion into a low power mode during sleep (i.e. kill power to it, and re-connect after resume).

## RFID Configuration:

Each RFID Mesa2 is pre-configured with a configuration. This configuration is stored as an SMBIOS key (similar to how the device serial number is stored). It can be obtained by reading the Chassis Version (see sample code). This configuration consists of three items.

- 1) **The region**, such as NA2, EU3, or AU (for North America, Europe, or Australia). This corresponds to ThingMagic's regions that are set using the following function.

```
RfidReader.ParamSet("/reader/region/id", configuredRegion);
```

- 2) **The maximum average power level.** The M6e Nano module is capable of running at 27 dBm, but due to regulatory reasons, you cannot operate at this power level continuously. For example, in North America, this power level is 20 dBm (2000 centi-dBm), and so if the RFID module is going to be operated continuously, 20 dBm is the highest power setting you can use. If, however, you are going to be scanning intermittently, you just need to ensure that the average power is kept below 20 dBm by preventing scans for a certain amount of "off time" following a scan (see "Scanning" section below).
- 3) **Disable Cellular.** If the disable cellular flag is set, then cellular communications are not allowed at the same time as RFID scanning. To scan for RFID, first you must disable cellular, then scan RFID, and then re-enable cellular.

Here are two example RFID Configurations:

- 1) NA2, 2000, 0 – This is configured for North America (using NA2), has a maximum average power level of 2000 centi-dBm (20 dBm), and does not need to disable cellular during RFID scans.
- 2) EU3, 2400, 1 – This is configured for Europe (using EU3), has a maximum average power level of 2400 centi-dBm (24 dBm), and needs to disable cellular during RFID scans.

## Software Components used:

The Mesa2 includes an expansion driver. This is how power is turned on or off to the RFID module; and cellular is disabled or re-enabled.

## More about disabling cellular:

There is a GPIO that connects the main processor to the cell module that, when asserted, shuts off the RF section of the cell module (basically puts it into airplane mode). If the cellular is disabled for a short enough time (perhaps less than 2 seconds), then the network has a good chance of remaining connected. Otherwise, the cellular network will disconnect, and will take roughly 15 seconds to reconnect after cellular is re-enabled.

## Source Code:

### Application Startup

When this application starts up, the onCreate() function hooks up to the expansion driver (appCreate()), checks for a valid expansion ID, reads in the Region, sets up some USB intent filters, and then turns on power. Upon power being turned on, the RFID will now enumerate on USB, and initialization continues in the BroadcastReceiver. Once the app has permission from the user, the remaining initialization is handed off to a thread (InitThread()). In this thread, communication is established.

### Handling Sleep/Resume

In onCreate(), we register for screen on/off broadcast notifications. Before the Mesa2 enters sleep, it is necessary to close the RfidReader and power down the RFID module. This means that after the Mesa2 resumes, that most of the startup code must be ran again.

### Scanning and handling “Off Time”

See the TriggerScan() function. Here, when we are ready to scan, we check to see if we are allowed to scan yet or not. If the last scan was more powerful than the maximum average power level allowed, we might have to wait for so much “off time” before we can allow a scan again. Next, if the Disable Cellular flag was set in the configuration, then we disable cellular before starting the scan. The scan time is timed so we can calculate how much off time we must enforce for next time. See also the calculateOffTime() function.