1.Enable PowerManagerService log.

public final class PowerManagerService extends SystemService

implements Watchdog.Monitor {

private static final String TAG = "PowerManagerService";

private static final boolean DEBUG = true;

private static final boolean DEBUG_SPEW = DEBUG && true;

…

2.Below is the call stack that PMS set a wakelock from userspace to kernel space, finally it will write the wakelock into the sysfs nodes. You can add logs inline to confirm whether the wakelock is set into the kernel space successfully.

/frameworks/base/services/core/java/com/android/server/power/PowerManagerService.java

acquireWakeLockInternal

-->updatePowerStateLocked

-->updateSuspendBlockerLocked

-->mWakeLockSuspendBlocker.acquire

-->nativeAcquireSuspendBlocker(mName)

/frameworks/base/services/core/jni/com_android_server_power_PowerManagerService.cpp

nativeAcquireSuspendBlocker

-->acquire_wake_lock

/hardware/libhardware_legacy/power/power.c

acquire_wake_lock

-->write(fd, id, strlen(id)) //the fd is /sys/power/wake_lock

3.In PowerManagerService construction, there are two PMS wakelocks:

```
    mWakeLockSuspendBlocker = createSuspendBlockerLocked("
    PowerManagerService.WakeLocks"); //for others

    mDisplaySuspendBlocker = createSuspendBlockerLocked("PowerManagerService.Display"); /
    /for LCD display
```

4.Be cautious, in PMS, below method will disable all PARTIAL WAKELOCK held from apps except those are in the whitelist when android enter doze idle mode. If you used dumpsys power, you can see the wakelock is marked as "Disabled".

```
private boolean setWakeLockDisabledStateLocked(WakeLock wakeLock) {

if ((wakeLock.mFlags & PowerManager.WAKE_LOCK_LEVEL_MASK)

== PowerManager.PARTIAL_WAKE_LOCK) {

boolean disabled = false;

if (mDeviceIdleMode) {

final int appid = UserHandle.getAppId(wakeLock.mOwnerUid);

// If we are in idle mode, we will ignore all partial wake locks that are

// for application uids that are not whitelisted.

if (appid >= Process.FIRST_APPLICATION_UID &&

Arrays.binarySearch(mDeviceIdleWhitelist, appid) < 0 &&

Arrays.binarySearch(mDeviceIdleTempWhitelist, appid) < 0 &&

mUidState.get(wakeLock.mOwnerUid,

ActivityManager.PROCESS_STATE_CACHED_EMPTY)

> ActivityManager.PROCESS_STATE_FOREGROUND_SERVICE) {

disabled = true;

}

}

if (wakeLock.mDisabled != disabled) {
```

```
wakeLock.mDisabled = disabled;

return true;

}

}

return false;

}
```