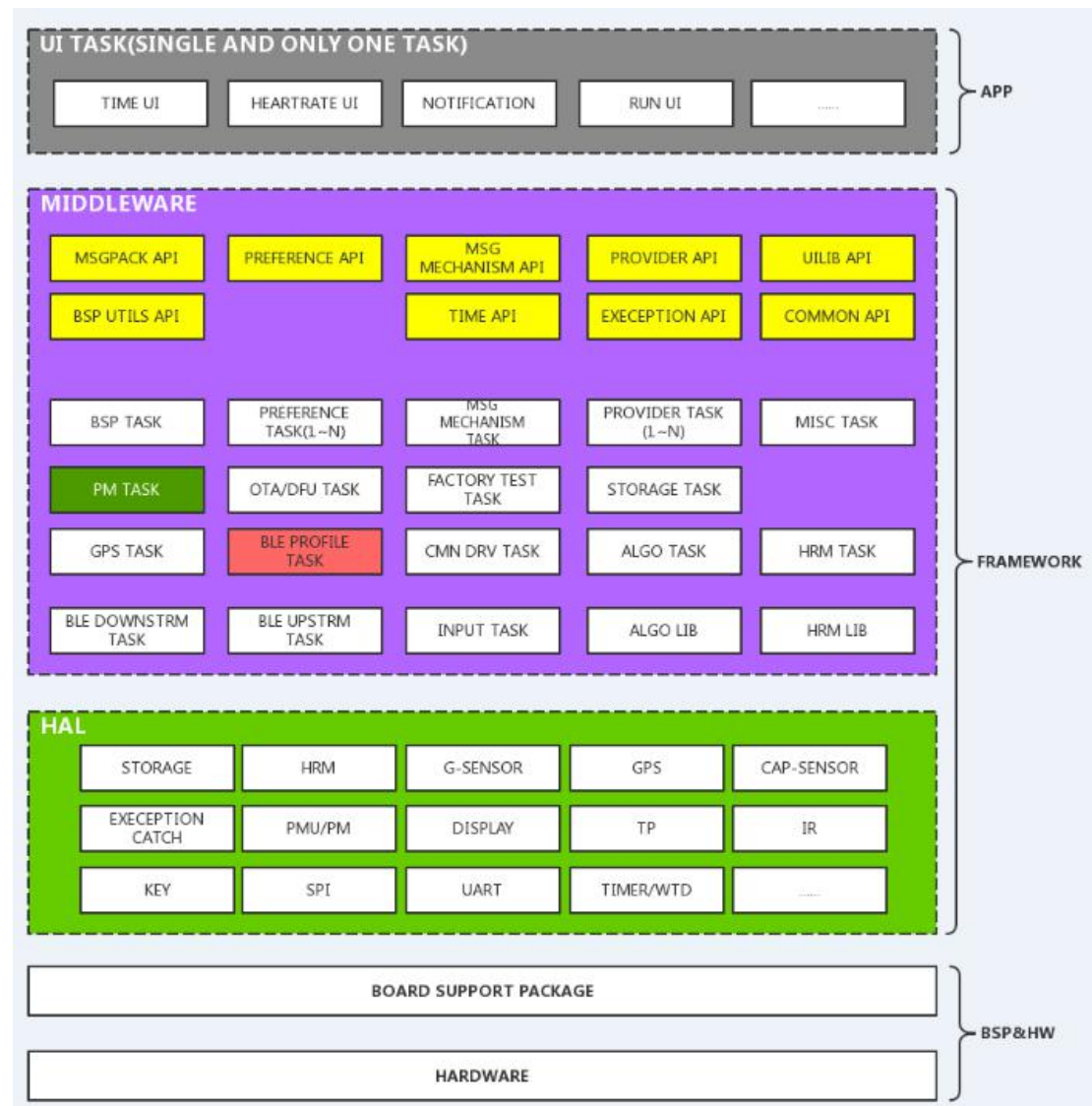


LANJING SYSTEM FRAMEWORK DESIGN

版本	修改日期	撰稿人	备注
V0.1	2017.12.11	张巨广	第一版框架设计稿

一 整体架构



二 BSP&HW

2.1 驱动所有内部组件和外设

2.2 抽象出所有外设和平台内部模块的标准接口

2.3 实现某些核心模块的功能（WATCHDOG）

三 FRAMEWORK

3.1 接口层

1. MSGPACK API

提供序列化和反序列化接口和实现，任务间通信和数据交换尽量都使用 MSGPACK。

2. UILIB

提供 UI 组件和事件框架。

3. PREFERENCE API

提供全局性设置、控制和配置信息，比如算法开关、用户信息、数据索引、启动项配置等；

4. PROVIDER API

提供数据访问接口，业务逻辑需要的数据通过该类 API 进行访问；

5. BSP UTILS API

提供对硬件外设访问和控制的接口。

6. COMMON API

提供通用的处理、操作和访问接口。

7. TIME API

提供对绝对时间和系统时间访问和操作的接口。

8. MSG MECHANISM API

提供系统和自定义事件收发的接口，含有两类接口：阻塞调用和费阻塞调用。

9. EXCEPTION API

被业务模块调用，用于记录系统异常发生时需要紧急备份的数据。

3.2 服务和实现层

1. PREFERENCE TASK

负责处理和组织自身和系统其它模块产生的数据，这些经过处理的数据或者缓存在 RAM 中，或者写入 FLASH 中。业务模块通过 PREFERENCE API 来访问和同步数据。

2. PROVIDER TASK

负责处理和组织自身或系统其它模块产生的数据，这些经过处理的数据或者缓存在 RAM 中，或者写入 FLASH 中。业务模块通过 PROVIDER API 来访问和同步数据。

3. BSP TASK

负责后台维护 BSP 硬件状态开关和控制的任務，与 BSP UTILS API 有部分对应关系。

4. MSG MECHANISM TASK

负责系统事件的异步路由处理，与 MSG MECHANISM API 有部分对应关系。

5. PM TASK

负责系统的电源工作状态（电量采集和处理、待机和运行、亮屏与灭屏等），数据和状态可以通过 PROVIDER API 或 MSG MECHANISM API 发给 PROVIDER 进一步处理。

6. OTA/DFU TASK

负责系统升级（固件包传输、DFU MODE 切换等）。

7. GPS TASK

负责 GPS NIMA 数据处理的轨迹里程优化，产生的结果数据通过 PROVIDER API 或 MSG MECHANISM API 发给相应的 PROVIDER 处理。

8. ALGO TASK

负责算法调度（按照一定频率 feed data）和结果收集，结果数据可以通过 PROVIDER API 或 MSG MECHANISM API 发给相应的 PROVIDER 做进一步处理。

9. HRM TASK

负责心率算法的调度和结果收集，结果数据可以通过 PROVIDER API 或 MSG MECHANISM API 发给相应的 PROVIDER 做进一步处理。可以考虑和 ALGO TASK 合并。

10. INPUT TASK

负责系统触摸和按键时间的收集、运算、判断和分发，比如长按，双击，划动等时间都在这个任务中处理。事件分发通过 MSG MECHANISM API 进行。

11. CMN DRV TASK

作为驱动程序的下半部（中断处理在中断上下文执行，后续的耗时操作统一在这个 TASK 上下文执行）。

12. MISC TASK

负责处理系统中没有严格区分领域的任务，以及后续额外添加的一些后台处理等。

13. BLE PROFILE TASK

负责 BLE 私有协议的组装、解析和路由，这是 BLE 私有协议实现的主战场。

14. BLE DOWNSTRM TASK

负责发送 BLE PROFILE TASK 组装好的私有协议包到手机端，采用串行化队列异步执行。

15. BLE UPSTRM TASK

负责接收手机端发过来的私有协议包，通过 MSG MECHANISM API 通知到 BLE PROFILE TASK 进行解析和路由，采用串行化队列异步执行。

16. FACTROY TEST TASK

完成产线测试功能，根据 ROM 配置决定是否启动。

17. STORAGE TASK

处理数据存储和查询。

四 UI

4.1 SINGLE UI TASK

应用 UI、通知 UI 和 INDICATOR UI 等调度时序和 OVERLAY 层级有一定复杂度，分到多个 TASK 处理，不容易控制，所以采用 SINGLE UI TASK 来统一集中调度。SINGLE UI TASK 负责处理页面切换、通知提醒、INDICATOR 切换、动画处理、UI 层级控制等。UI 开发工作按照业务领域高度模块化，多人并行开发。

4.2 数据和事件

UI 所需要的数据来源有两种，一种是通过 MSG MECHANISM 被动通知，另一种是通过 PROVIDER API 或 PREFERENCE API 主动获取；事件通知统一由 MSG MECHANISM 提供，SINGLE UI TASK 被动监听，收到事件后再分发给各业务 UI 呈现。