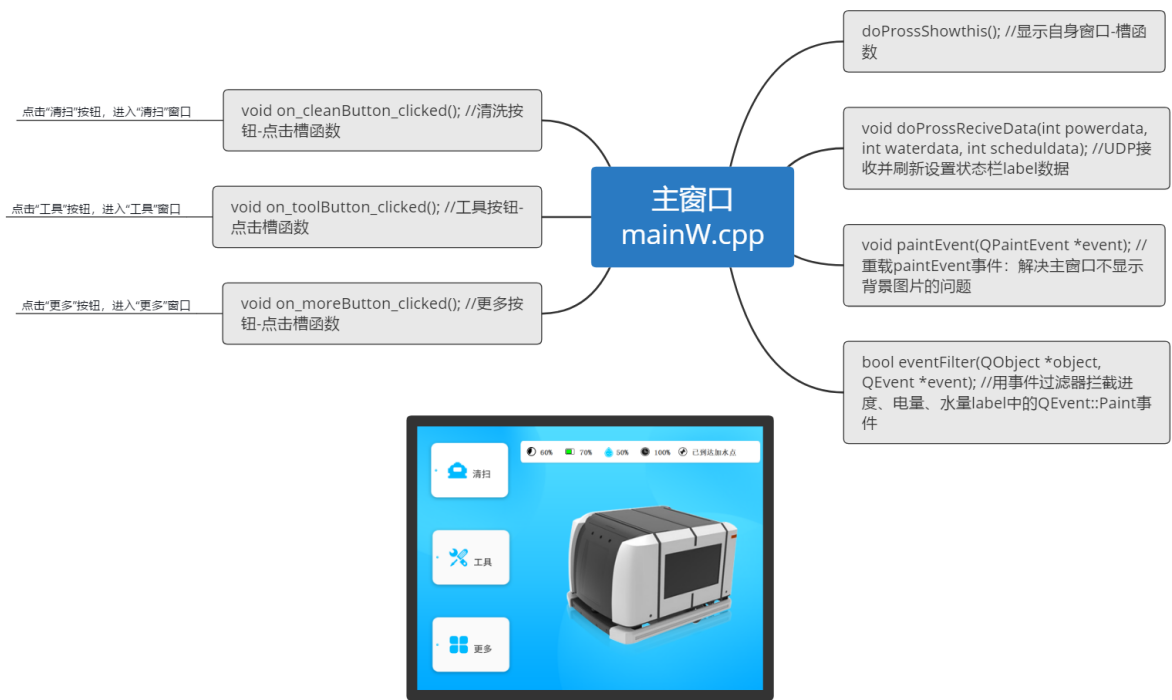


一、主窗口

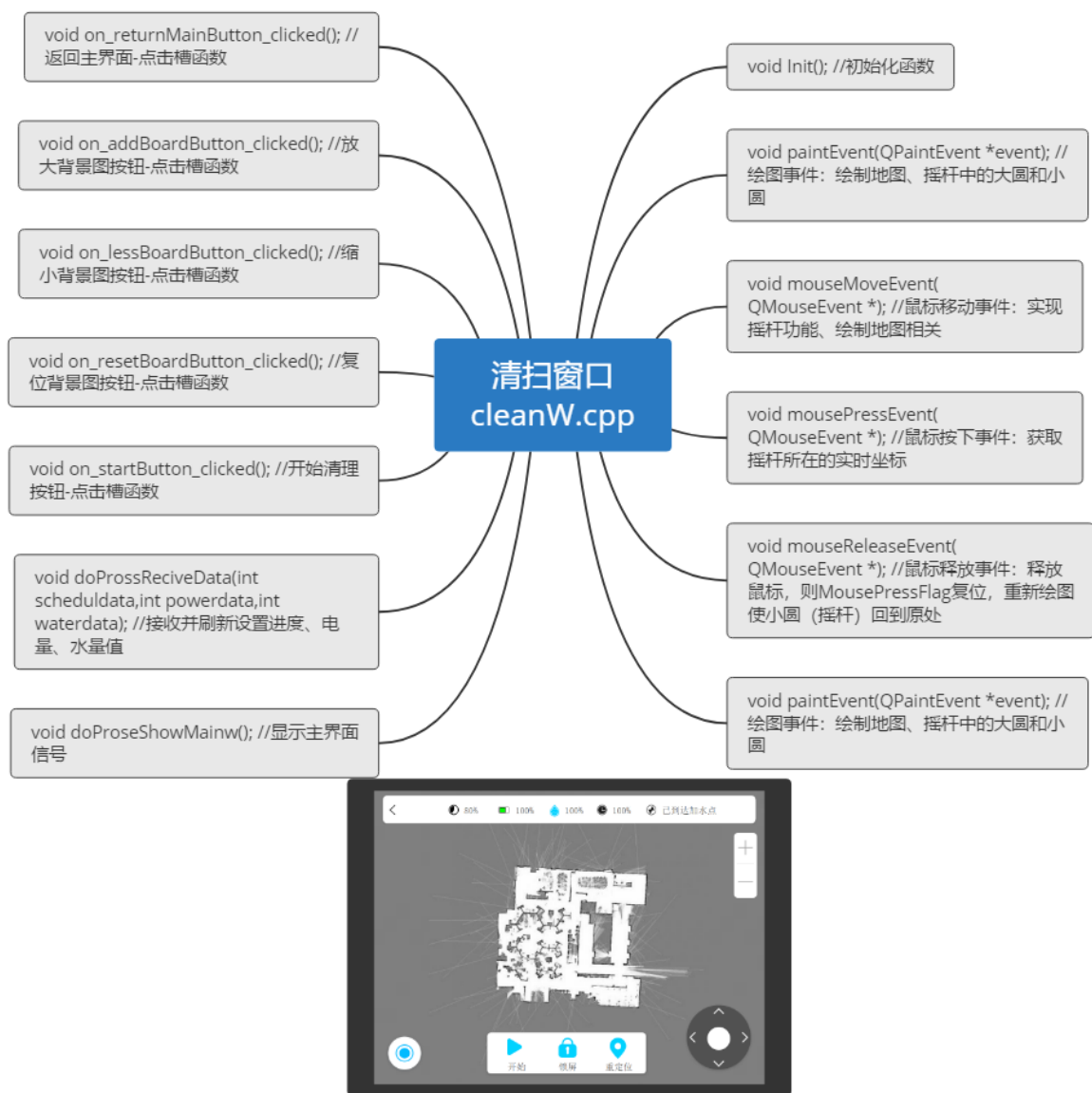
下面是"主窗口"的思维导图：



在经过"开机加载窗口"后，则开始显示主窗口(已经解除锁屏状态后)，左侧是3个菜单按钮，可以选择进入3个子菜单窗口，上方是扫地机器人的各状态值，每隔一段时间通过 UDP+串口 读取扫地机器人的状态值，例如剩余电量值、剩余水量值等等。

二、清扫窗口

下面是"清扫窗口"的思维导图：



在"主窗口"中点击"清理"按钮, 则进入"清理窗口", 在该窗口中显示要清扫的房间地图, 同时扫地机器人执行清扫工作。

可以点击右侧的"放大地图"和"缩小地图"按钮, 来放大和缩小地图; 另外点击左下角的"复位地图"按钮可以复位地图; 右下角的是"摇杆"。

三、工具窗口

下面是"工具窗口"的思维导图:



在"工具窗口"中选择 风机开启/关机、吸水耙上升/下降、滚刷开启/关闭、滚刷上升/下降、进入广告窗口、进入设置滚刷速度窗口。

四、更多窗口

下面是"更多窗口"的思维导图：



在"更多窗口"中使用 生成路径、切换地图包、锁屏设置 等功能。

五、修改密码窗口

下面是"修改密码窗口"的思维导图：



功能：

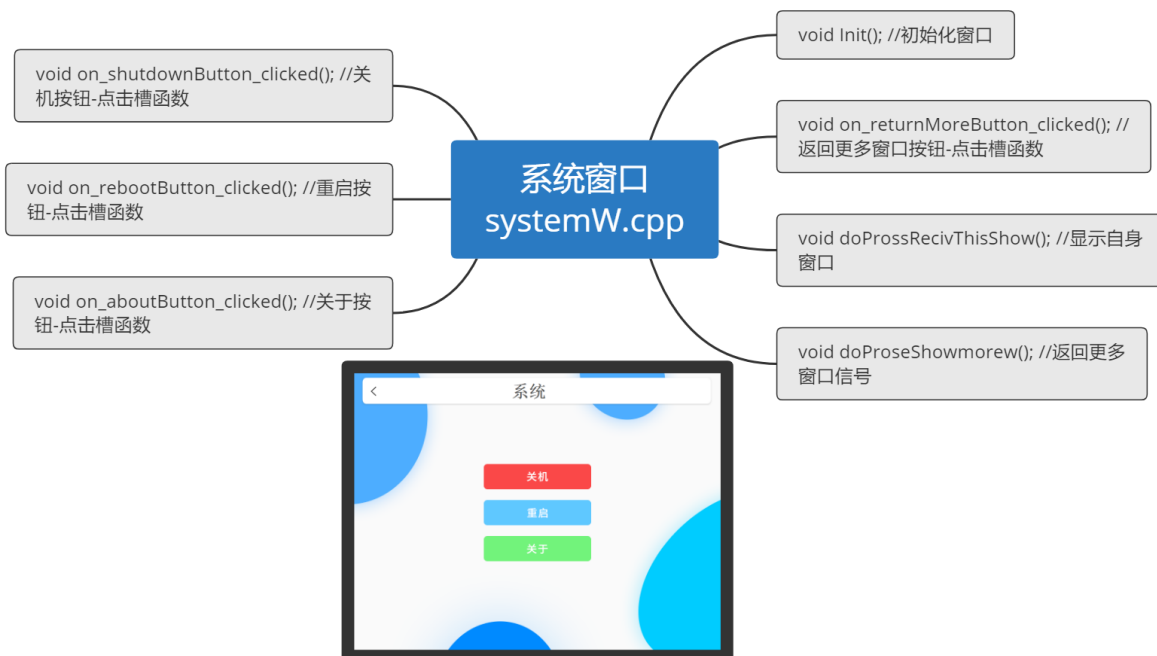
在"修改密码窗口"窗口中，点击任意一行密码栏，会弹出小键盘，点击小键盘中的数字输入密码；最开始进入该界面时，按钮为灰色不可点击状态，三行密码栏都输入6位数字后，按钮变亮为可点击状态，点击"确定"按钮来修改密码。

分析：

- `void init();`
 - 设置窗口无边框且窗口显示在最顶层；
 - 显示默认提示；
 - 三行密码栏安装事件过滤器；
 - 通过正则表达式设置三行"密码输入栏"，只能输入6位数字。
- `bool eventFilter(QObject *watched, QEvent *event);`
 - 当焦点在原密码栏，设置原密码栏的样式为"蓝色边框"，第一次点击"确认密码栏"时清除密码栏文本，设置"确认密码栏"为密码格式，其它两行密码栏类似；
 - 判断三行密码栏的位数是否均是6位，是则按钮变为淡蓝色，且设置确定按钮可用。
- `void showDefaultTip();`
 - 设置三行密码栏的样式都为"灰色边框"；
 - 设置三行密码栏的文本都为"默认文本"；
 - 确定按钮变为灰色，且设置确定按钮不可用。
- `on_returnLockButton_clicked();`
 - emit `doProseShowlockscreen()` 信号，关闭本窗口，返回显示上一级窗口 - 锁屏设置窗口；
 - 另外在上一级窗口中：
`connect(mychangpassword, SIGNAL(doProseShowlockscreen()), this, SLOT(doProseShowthis()));`，来实现返回上一级窗口的功能。
- `void on_okButton_clicked();`
 - 首先在配置文本中读取原密码；
 - 判断输入的新密码(确认密码)与原密码是否一致，一致则写入新密码到配置文本中，并弹出提示框提示用户修改密码成功，否则则警告用户输入错误。
- `void on_okButton_pressed();`
 - 由于前面使用了 `setStyleSheet()` 函数，QtDesigner样式表失效，需要这里再使用 `setStyleSheet()` 函数。

六、系统窗口

下面是"系统窗口"的思维导图：



功能:

在"系统窗口"中, 点击"关机"和"重启"按钮, 会弹出提示框, 点击提示框中的"确定"按钮, 若是Windows系统则会关闭和重启应用软件, 若是UBuntu系统则会关机 and 重启; 而点击"关于"按钮, 则会弹出"关于窗口", 展示"关于信息"。

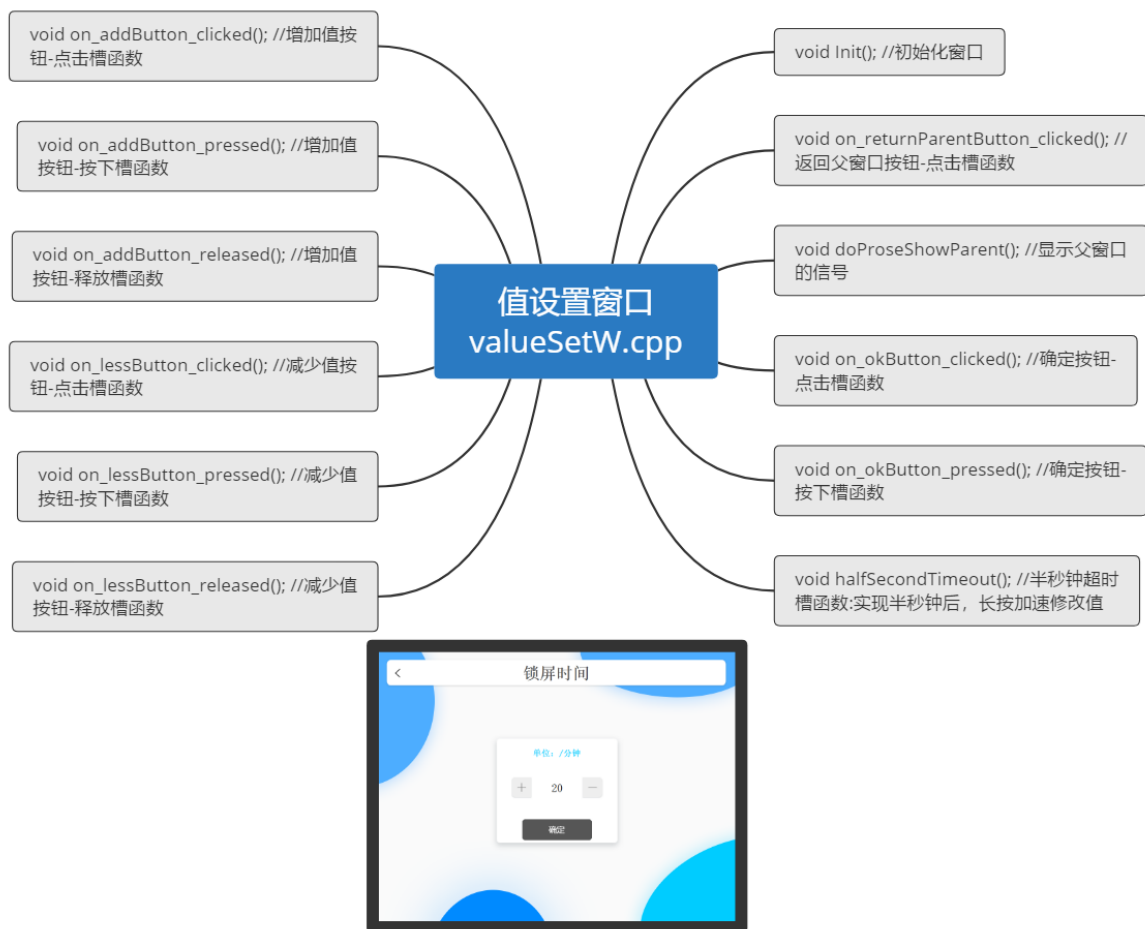
分析:

- `void Init();`
 - 初始化"提示框"窗口实例;
 - 初始化QSignalMapper实例, 将两个按钮的点击信号和QSignalMapper实例的map()槽函数关联, "xx按钮"字符串作为实参传递;
 - 然后将QSignalMapper实例的mapped信号与"提示框"窗口实例的recvButton槽函数关联, 传递"xx按钮"字符串;
- `void on_shutdownButton_clicked();`
 - 使用 `setTitle2Info()` 函数修改弹出"提示框"的标题和提示信息, 并显示"提示框"窗口。
- `void on_aboutButton_clicked();`
 - 创建"关于"窗口, 显示该窗口, 并连接"关于"窗口的返回上一级窗口的信号与槽。

当按下"关机按钮"时, 可以传递"关机按钮"字符串给"提示框"窗口实例, 使得"提示框"窗口实例根据"字符串"知道是"关机按钮"按下, 而不是"重启按钮", 然后在其 `on_okButton_clicked` 槽函数中执行"关机"功能。按下"重启按钮"同理, 这里不再赘述。

七、值设置窗口

下面是"值设置窗口"的思维导图:



功能:

在"值设置窗口"中, 进行值的设置, 目前用来设置"锁屏时间"和"滚刷速度", 这两者用同一个窗口设置值; 长按超过半秒, 可以加速修改值, 另外值也有范围限制。

分析:

- `void Init();`
 - "增加值按钮按下定时器"的超时信号, 与"增加值按钮-点击槽函数"绑定;
 - "减小值按钮按下定时器"的超时信号, 与"减小值按钮-点击槽函数"绑定;
 - "按钮按下半秒钟定时器"的超时信号, 与"halfSecondTimeout槽函数"绑定;
 - 初始设置确定按钮不可用;
 - 读取配置文本中的配置值, 并相应设置label值。
- `void on_addButton_clicked();`
 - 增加值, 确定按钮设置高亮, 且设置确定按钮可用。
- `void on_addButton_pressed();`
 - 增加/减少值的标志位设置为true;
 - 按下按钮, 则按钮按下半秒钟定时器开始计时。
- `void on_addButton_released();`
 - 按钮释放后, 则"按钮按下半秒钟定时器"和"增加值按钮按下定时器"停止计时。
- `void halfSecondTimeout();`
 - 关闭"按钮按下半秒钟定时器";
 - 根据"增加/减少值的标志位", 选择开启"增加值按钮按下定时器"还是"减小值按钮按下定时器"。

按下"增加值按钮", 开启"按钮按下半秒钟定时器", 超过半秒后进入 `halfSecondTimeout()` 函数, 则会开启"增加值按钮按下定时器"; 当触发"增加值按钮按下定时器"的超时信号后, 会调用"增加值按钮-点击槽函数"增加值; 按下"减少值按钮"同理, 这里不再赘述。

八、
