X型咖啡机一次可以制作一壶咖啡。用户把过滤器放置在支架上，在其中装入研磨好的咖啡,然后把支架推入。接着，用户向水箱倒入足够的水并按下冲煮(Brew)按钮。加热器加热水一直加热到沸腾。不断产生的水蒸气压使水洒在咖啡粉末上,凝结的水滴通过过滤器流入到咖啡壶中。咖啡壶由一个保温盘进行长期保温，仅当壶中有咖啡时，保温盘才进行工作。如果在水还在向咖啡粉喷洒时从保温盘上拿走咖啡壶，水流就会停止，这样煮好的咖啡就不会溅在保温盘上。以下是需要监控的硬件设备。

加热器 (Boiler Heater)：加热可以开启和关闭。  
保温器 (Warmer)：加热可以开启和关闭。  
保温盘 (Warmer Plate)传感器：它有3个状态——warmerEmpty（咖啡壶不在保温盘上）、potEmpty（咖啡壶在保温盘上但没有咖啡）和potNotEmpty（咖啡壶在保温盘上且有咖啡）。  
加热器 (Boiler)传感器：用来判断水箱是否有水。它有两个状态:　boilerEmpty、boilerNotEmpty。煮咖啡过程由boilerEmpty作为结束的节点。  
冲煮按钮 (Brew Button)：这个瞬时按钮启动冲煮流程。用户开始煮咖啡时如果水箱无水，或者上一次煮咖啡过程没有结束，则不会重新开始煮咖啡。  
指示灯 (Indicator)：冲煮开始后熄灭，直到当冲煮流程结束时再次亮起，表示咖啡已经煮好。  
减压阀门 (Relief Valve)：在开启时可以降低加热器中的压力。压力降低会阻止水流向过滤器。该阀门可以开启和关闭。  
X型专用咖啡机的硬件已经设计完成，硬件工程师为我们提供了低层的API，如下:

namespace CoffeeMaker {  
enum WarmerPlateStatus {  
    WARMER\_EMPTY, POT\_EMPTY, POT\_NOT\_EMPTY  
};  
enum BoilerStatus {  
    EMPTY, NOT\_EMPTY  
};  
enum BrewButtonStatus {  
    NOT\_PUSHED, PUSHED  
};  
enum BoilerHeaterState {  
    OFF, ON  
};  
enum WarmerState {  
    OFF, ON  
};  
enum IndicatorState {  
    OFF, ON  
};  
enum ReliefValveState {  
    CLOSED, OPEN  
};  
/\*  
 \* This function returns the status of the warmer-plate  
 \* sensor. This sensor detects the presence of the pot  
 \* and whether it has coffee in it.  
 \*/  
WarmerPlateStatus GetWarmerPlateStatus();  
/\*  
 \* This function returns the status of the boiler switch.  
 \* The boiler switch is a float switch that detects if  
 \* there is more than 1/2 cup of water in the boiler.  
 \*/  
BoilerStatus GetBoilerStatus();  
/\*  
 \* This function returns the status of the brew button.  
 \* The brew button is a momentary switch that remembers  
 \* its state. Each call to this function returns the  
 \* remembered state and then resets that state to  
 \* NOT\_PUSHED.  
 \*  
 \* Thus, even if this function is polled at a very slow  
 \* rate, it will still detect when the brew button is  
 \* pushed.  
 \*/  
BrewButtonStatus GetBrewButtonStatus();  
/\*  
 \* This function turns the heating element in the boiler  
 \* on or off.  
 \*/  
void SetBoilerHeaterState(BoilerHeaterState s);  
/\*  
 \* This function turns the heating element in the warmer  
 \* plate on or off.  
 \*/  
void SetWarmerState(WarmerState s);  
/\*  
 \* This function turns the indicator light on or off.  
 \* The indicator light should be turned on at the end  
 \* of the brewing cycle. It should be turned off when  
 \* the user presses the brew button.  
 \*/  
void SetIndicatorState(IndicatorState s);  
/\*  
 \* This function opens and closes the pressure-relief  
 \* valve. When this valve is closed, steam pressure in  
 \* the boiler will force hot water to spray out over  
 \* the coffee filter. When the valve is open, the steam  
 \* in the boiler escapes into the environment, and the  
 \* water in the boiler will not spray out over the filter.  
 \*/  
void SetReliefValveState(ReliefValveState s);  
}

现在请使用底层API用C++编写一个咖啡机程序。 并且提供必要的流程图等文档说明。

**答：**

**一、按照事件-状态模型来设计代码**

**1.1事件**有：

E1--按下冲煮按键

**1.2状态**有：

传感器读取类状态：

SA1{保温盘状态WarmerPlateStatus}

SB1{加热器水箱是否有水BoilerStatus}

SC{冲煮按钮状态BrewButtonStatus}

设置驱动器状态：

X1[设置减压阀状态ReliefValveState]

X2[设置指示灯状态IndicatorState]

X3[设置保温器状态WarmerState]

X4[设置加热器是否加热BoilerHeaterState]

**二、流程图如下所示：**

