

Tasks<<enum>>

NoIndicator: 0
Resting: 1
Kneading: 2
Rising: 3
Baking: 4
Done: 5

States<<enum>>

Standby
...
(must be filled in after making your state diagram)

Events<<enum>>

NoEventOccurred
MenuBtnPressed
MenuBtnLongPressed
TimerUpBtnPressed
TimerDownBtnPressed
StartBtnPressed
OvenDone
TimerTimeout

IOvenSimulator<<interface>>

SetTemperature(temp: int)

Oven

Oven()
AddTimer(timer: ITimer&)

IOven<<interface>>

StartRise(timeInMinutes: int)
StartBake(timeInMinutes: int)
IsOn(): bool
GetTemperature(): int
Cancel()

ITimerTimeout<<interface>>

TimerTimeout()

Timer

Timer(timerTimeout: ITimerTimeout)

ITimer<<interface>>

Set(time: uint64_t)
Cancel()

IUserActions<<interface>>

MenuPressed()
MenuLongPressed()
TimerUpPressed()
TimerDownPressed()
StartPressed()

EventGenerator

EventGenerator(IOven& oven)
GetEvent(): Event

IEventGenerator<<interface>>

GetEvent(): Event

BreadBaker

BreadBaker(...) // you must fill this in yourself!

Run()

IKneadMotor<<interface>>

TurnLeft()
TurnRight()
Stop()

KneadMotor

IYeastTray<<interface>>

Drop(uint64_t time)
Cancel()

YeastTray

IExtraIngredientTray<<interface>>

Drop(uint64_t time)
Cancel()

ExtraIngredientTray

IDisplay<<interface>>

SetCurrentTask(task: CurrentTask)
SetMenuNumber(number: int)
SetTime(hour: int, min: int)
DisplayOff()
(first 3 methods automatically switch display on)

Display

IStartButtonLed<<interface>>

LedOn()
LedOff()

StartButtonLed

