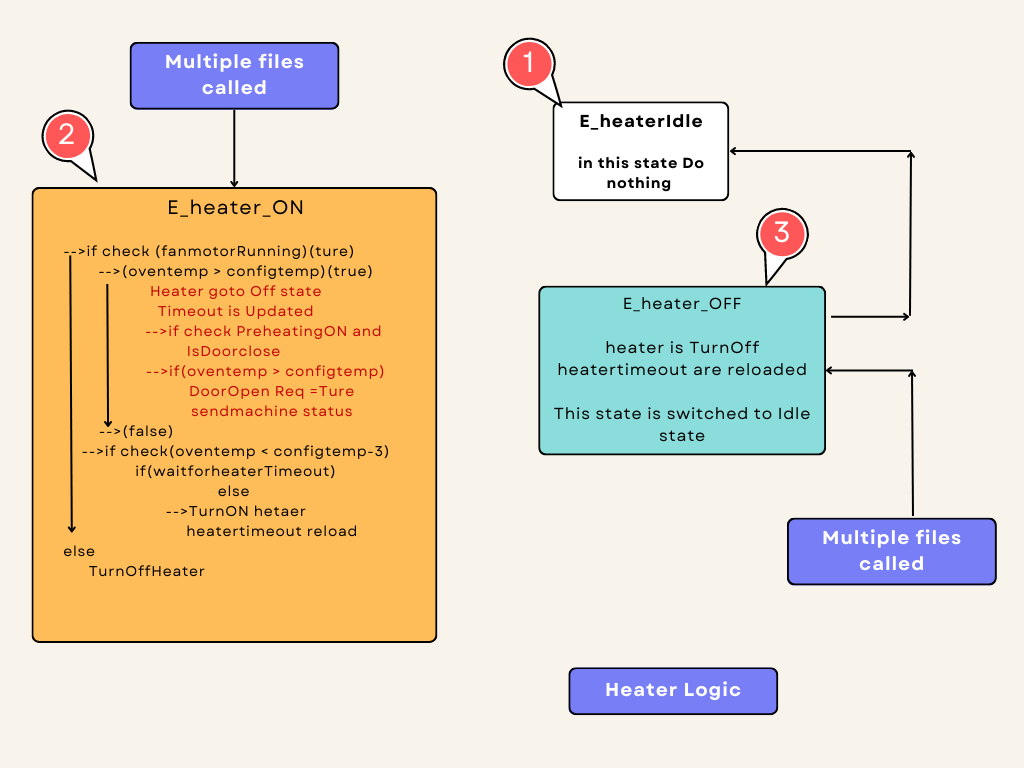
**Heater\_Logic**



**Using Macros**

* **HEATER\_VALIDATE\_COUNTER**

🡪 This macro is used to validate the Heater is ON or OFF.

* **HEATER\_OFF\_WAIT\_TIMEOUT**

🡪 This macro is used to give some delay to change the will-go heater off state.

* **FAN\_MOTOR\_NOT\_WORKING,**
* **HEATER\_NOT\_TURNING\_OFF ,**
* **HEATER\_NOT\_WORKING**

🡪 These macros are used to report the error in the Diagnostic Report.

**Using variables**

* **heaterControl**

🡪This variable is used to control the states of the switch method.

* **WaitForHeaterOn**

🡪 This variable is represented by the macro of the HEATER OFF WAIT\_TIMEOUT.

* **HeaterTurnOnTimeOutCounter**

🡪 This variable is represented by the macro of the HEATER\_VALIDATE\_COUNTER. it tells the ON\_time of the heater.

* **HeaterTurnOffTimeOutCounter**

🡪 This variable is represented by the macro of the HEATER\_VALIDATE\_COUNTER. it tells the OFF\_time of the heater.

**Configured temperature**

🡪 This variable holds the current step of config Temp in the Oven APP. This variable is handled in the Cookingconfig file.

* **OvenTemprature**

🡪 This variable to holds the currently inside the Oven Temperature. this Variable is Handled in the SPI\_Sensor file.

**Flow Of states**

* **Heater Idle**
* **Heater\_ON**
* **Heater\_OFF**

1. **Heater\_Idle**

* In this state heater remains idle, and after that, we will switch to Heater\_ON state.

1. **Heater\_ON**

* First, will check some conditions, If it’s true go inside the block and turnOn heater and set the heater-on-off timer.
  + **Inside the Conditions**

**🡪 (IsHeaterTurnedOn())**

**🡪 (IsFanMotorRunning())**

**🡪(ovenTemprature > configuredTemprature)**

**🡪 (IsPreheating())**

**🡪(ovenTemprature < (configuredTemprature - 3))**

**Conditions Explanation**

**🡪 (IsHeaterTurnedOn())**

**🡪** This condition is to check whether the heater is ON, if It is True and further action is done. If it's false, this is directly sent to (IsHeaterTurnedOff()) and it is Heater Off and updated by the Timeout\_values.

**🡪 (IsFanMotorRunning())**

**🡪** This condition is check finds whether the motor is running or not, after the motor is turned on, the Heater will turn ON.

🡪 Suppose if it's not true, directly this goes to the heater\_turned off state.

**🡪 (ovenTemprature > configuredTemprature)**

**🡪**This condition is tells, If the oventemp is greater than the Configtemp the heater status is changed to a turnoff state.

**🡪 (WaitForHeaterOn > 0)**

* This logic will give some delay to turn OFF the heater.
* **if(HeaterTurnOffTimeOutCounter > 0)**

**🡪**This logic is to give some Time to check the Heater is OFF or NOT. And the report sent it to HMI.

🡪 **(IsPreheating())**

🡪 if this condition checks that the preheating is enabled in Cooking Mode If it’s true, and check the door is closed state, the machineStatus\_DOR bit requests to set and send the status to HMI , Request to open The Door.

🡪**OpenRequest** Flag Is set in True, This is Handle to Door Logic File, This is represented by The Oven is in Preheat mode.

**🡪(ovenTemprature < (configuredTemprature - 3))**

**🡪** This condition is checks for whether ovenTemp is low compared with 3 points minus the CongfigTemp in HMI, if it's true, heater will be turned On. then Timeout\_values are updated.

**(HeaterTurnOnTimeOutCounter > 0)**

**If this condition is true,then the HeaterTurnOnTimeOutCounter variable will be decremented for every function call.**

This logic is used to find the Heater is Working Or Not. After that decrement is complete, that time also Oven has not reached the config temperature, it give the report to the HMI.

1. **Heater\_OFF**

**🡪** In the state, it goes to turn off the heater, then the heaterTurnOnWaitTime\_values are updated. WaitForHeaterOn, HeaterTurnOnTimeOutCounter, and HeaterTurnOffTimeOutCounter variables are updated with macros.

🡪And finally, this state is switching to the Idle state.