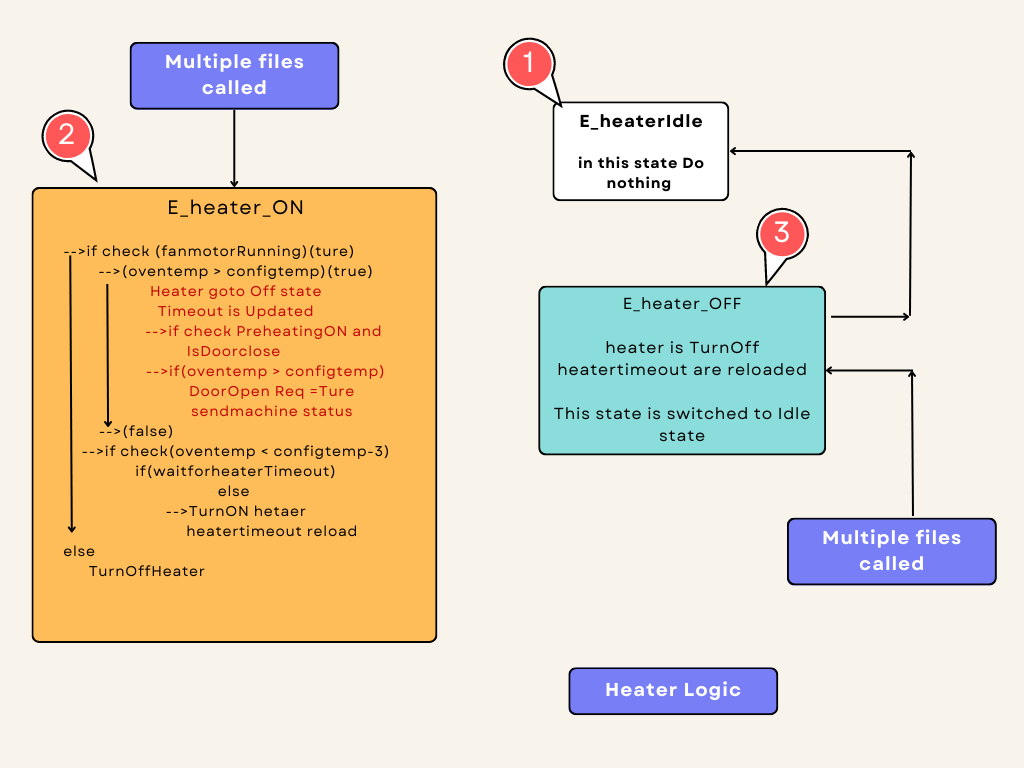
**Heater\_Logic**



**Using Macros**

* **HEATER\_VALIDATE\_COUNTER**

🡪This macro is used for the Heater On/Off Time.

* **HEATER\_ON\_WAIT\_TIMEOUT**

🡪This macro is used for HeaterOn waiting Time.

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

🡪 These macros are used for Diagnostic Report.

**Using variables**

* **heaterControl**

🡪This varible is used to control the Machine states.

* **WaitForHeaterOn**

🡪 This variable is used for HeaterOn waiting Time. it's used to update by this macro HEATER\_ON\_WAIT\_TIMEOUT

* **HeaterTurnOnTimeOutCounter**

🡪 This varible is used for On\_Time of the heater.

* **HeaterTurnOffTimeOutCounter**

🡪 This varible is used for Off\_Time of the heater.

* **configuredTemprature**

🡪 This variable is used to take values From the HMI Temp\_Config value.

* **ovenTemprature**

🡪 This variable is used to take a value from OvenTemp using Max6675 in spi\_sensor.

**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 turnOn.

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

**🡪 (ovenTemprature > configuredTemprature)**

**🡪**This condition is tells, If the oventemp is greater than the Configtemp the heater status is changed to a turnoff state and also set the heater-on-off timer. And the report send it.

🡪 **(IsPreheating())**

🡪 if this condition is checks that the preheating is enabled in Cooking Mode If it’s true, and the door is in the closed state, machineStatus\_DOR bit requests to open state.

**🡪**Suppose it's false it goto next block(else). Below the else block

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

**🡪** This condition is checks for whether ovenTemp is low compared with CongfigTemp in HMI, if it's true, Get some seconds and after heater will be turned On. then Timeout\_values are updated.

**🡪 (WaitForHeaterOn > 0)**

If this condition is true then WaitForHeaterOn variable is decremented for every function call.

Suppose if it is false,then TURN\_ON\_HEATER() macro will be called and the HeaterTurnOffTimeOutCounter will be updated with the HEATER\_VALIDATE\_COUNTER macro. There is one more check under this condition.

**(HeaterTurnOnTimeOutCounter > 0)**

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

Suppose if it is false and the oven temperature less than configured temperature,then results in a diag event reporting and HeaterTurnOnTimeOutCounte variable will be updated with HEATER\_VALIDATE\_COUNTER macro.

1. **Heater\_OFF**

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

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