**Cooking\_Config\_setting**

**Variables**

**androidData\_ST androidData 🡪** This ST is used to store the android data.

**recipeStepMode\_EN recipeStepMode 🡪** This Enum is used to Control the Cooking Modes.

**machineStaus\_EN machine\_Status 🡪** This is Represented by the Machine status,

**maximumTemprature 🡪** This Variable is used to Store the Max Temp of the Default value.

**runTimeIn100Millis 🡪**

**configuredTemprature 🡪** This variable is used to store the Current Step of the Configured Temp.

**prevConfiguredTemprature 🡪** This Variable is used to Store the Current step temp, it Compared between the next step Configured Temp.

**enableMultiTrayMode 🡪** This Flag is used to Control the Multi-Tray mode and Normal cooking mode.

**androidDataRxCbk()**

**🡪**This Function Is represented by the Android data received Every 5 seconds.

🡪The data is Stored corresponding ST.  
🡪The Sync Flag indicates synchronization between the uC and HMI.

**configureTheCookingMode()**

🡪This Function is executed based on configured the cooking Mode and motor speed, temperature, cooking run time and steam level of configuration cooking of the current step.

recipeStepConfig\_ST \*rSconfig 🡪 This Pointer variable is used to select the mode and check preheat and check total steps of cooking

This function executes depending on the given cooking mode.

**🡪 Undefined Mode:**

🡪This mode represented by the cooking mode is not Configured, so It’s reported to HMI.

**🡪Hot air Mode:**

🡪This is one of the Cooking Mode, it’s represented by without Steam cooking.

🡪 if(!(recipeSteps[currentStepIndex].disable)): Checks if the current step is not disabled. If the step is disabled, it skips the entire logic block for this step.

**🡪 if(currentStepIndex != (recipeSteps[currentStepIndex].stepNumber - 1))**, This check is used to Report to the HMI, to Receive the Invalid Step Number.

**Check the Preheating Mode:**

**🡪 if(IsPreheating() && (rSconfig->preHeat == TRUE))**, This check verify the Preheating mode Is enabled or not. suppose it’s enabled, the Preheating FanSpeed is configured as the update motor speed(). If it is false, to execute the normal fan speed, asthe update motor speed().

🡪 The Previous Configured temp is updated by the Configured temperature, For comparison of the next step Configured temperature.

🡪 The Configtemperature variable and MaximumTempreature Variable are updated by The maximum Temperature and Current Step temperature.

**🡪 if(configuredTemprature > maximumTemprature)**, This check is represented by the Report to HMI if the configured temp is greater than the Maximum Temp.

🡪The runTimeIn100Millis variable is updated by current step Configured cooking time, this time is converted to a 100ms task.

**🡪 if(runTimeIn100Millis == 0)**, This check is represented by the Multi-tray mode is enabled, if the runtime is not configured, it means the Multi-tray is enabled.

**🡪Steam Mode:**

🡪This is one of the Cooking Mode, it’s represented by with Steam.

🡪 if(!(recipeSteps[currentStepIndex].disable)): Checks if the current step is not disabled. If the step is disabled, it skips the entire logic block for this step.

🡪updateSteamLevel(recipeSteps[currentStepIndex].steamLevel,&CS\_RecipeSettings[STEAM\_RECIPE\_INDEX]); This functions is represent by updated the Steam level of the Steam mode.(Like low, medium, high and on time and off time).

**🡪Check and Activate Steam:**

Checks if the oven is not in preheating mode (!IsPreheating()) and not in idle cooking mode (!IsCookingIdle()).

If the conditions are met, it checks if the current step is the first step (currentStepIndex == 0). If true, it turns on the steam inlet (assumed by TurnOnSteam()); otherwise, it activates the steam (SteamOn()).

🡪 TurnOnSteam(),This function is doing Step On and off the Steam at a certain time, after that it will switch to the cyclic method.

🡪 SteamOn(), This function is doing a cyclic method with corresponding on and off time.

**Check the Preheating Mode:**

**🡪 if(IsPreheating() && (rSconfig->preHeat == TRUE))**, This check verify the Preheating mode Is enabled or not. suppose it’s enabled, the Preheating FanSpeed is configured as the update motor speed(). If it is false, to execute the normal fan speed, asthe update motor speed().

🡪 The Previous Configured temp is updated by the Configured temperature, For comparison of the next step Configured temperature.

🡪 The Configtemperature variable and MaximumTempreature Variable are updated by The maximum Temperature and Current Step temperature.

**🡪 if(configuredTemprature > maximumTemprature)**, This check is represented by the Report to HMI if the configured temp is greater than the Maximum Temp.

🡪The runTimeIn100Millis variable is updated by current step Configured cooking time, this time is converted to a 100ms task.

**🡪 if(runTimeIn100Millis == 0)**, This check is represented by the Multi-tray mode is enabled, if the runtime is not configured, it means the Multi-tray is enabled.

**🡪Combination Mode:**

🡪This is one of the Cooking Mode, it’s represented by with combination.

🡪 if(!(recipeSteps[currentStepIndex].disable)): Checks if the current step is not disabled. If the step is disabled, it skips the entire logic block for this step.

🡪updateSteamLevel(recipeSteps[currentStepIndex].steamLevel,&CS\_RecipeSettings[STEAM\_RECIPE\_INDEX]); This functions is represent by updated the Steam level of the Steam mode.(Like low, medium, high and on time and off time).

**🡪Check and Activate Steam:**

Checks if the oven is not in preheating mode (!IsPreheating()) and not in idle cooking mode (!IsCookingIdle()).

If the conditions are met, it checks if the current step is the first step (currentStepIndex == 0). If true, it turns on the steam inlet (assumed by TurnOnSteam()); otherwise, it activates the steam (SteamOn()).

🡪 TurnOnSteam(),This function is doing Step On and off the Steam at a certain time, after that it will switch to the cyclic method.

🡪 SteamOn(), This function is doing a cyclic method with corresponding on and off time.

**Check the Preheating Mode:**

**🡪 if(IsPreheating() && (rSconfig->preHeat == TRUE))**, This check verify the Preheating mode Is enabled or not. suppose it’s enabled, the Preheating FanSpeed is configured as the update motor speed(). If it is false, to execute the normal fan speed, asthe update motor speed().

🡪 The Previous Configured temp is updated by the Configured temperature, For comparison of the next step Configured temperature.

🡪 The Configtemperature variable and MaximumTempreature Variable are updated by The maximum Temperature and Current Step temperature.

**🡪 if(configuredTemprature > maximumTemprature)**, This check is represented by the Report to HMI if the configured temp is greater than the Maximum Temp.

🡪The runTimeIn100Millis variable is updated by current step Configured cooking time, this time is converted to a 100ms task.

**🡪 if(runTimeIn100Millis == 0)**, This check is represented by the Multi-tray mode is enabled, if the runtime is not configured, it means the Multi-tray is enabled.

**CookingConfigCheck()**

**🡪** This function verifies the all the steps are configured correctly or not.

🡪The CookingConfigCheck function iterates through the recipe steps based on the specified cooking mode.

🡪It checks if each step is disabled, and if not, it verifies whether the current step index matches the expected step number. If any discrepancies are found, it reports relevant diagnostic events and sets flags, and the final retval is set to FALSE.

🡪The debugging output includes the value of retval for further analysis.

**RecipeSettingsRxCbkCheck()**

🡪The RecipeSettingsRxCbkCheck function checks the number of steps and the value of RCF.Byte based on predefined conditions for different cases (1 to 6 steps).

🡪If the conditions are met, it sets retval to TRUE and clears relevant diagnostic flags. If the conditions are not met, it sets RCF.Byte to 0, sets retval to FALSE, and calls CookingDR to report diagnostics.

🡪 If the number of steps is not within the expected range (1 to 6), it logs an "InvalidStep" message. The final boolean value is then returned.