**M2M Challenge 1: Oven simulation.**



A computer screen shot of a computer

Description automatically generated

*By: Johnson Domacasse (#4471709)*

*20 February 2024*

# Introduction:

The purpose of this challenge is to get the user acquainted with the CODESYS program. This is done by giving the user a pre-made simulation of a pizza oven. The tasks the user has are to deconstruct the simulations state and implement these into a state machine diagram. The user must then add an additional function with the feature of changing between two different time stamps for the countdown. Currently the simulation has a set timer of 10000 milliseconds. An additional timer of 16000 milliseconds will be added. If the user wishes, they can add any additional features while maintaining the already existing features.

# State machine diagram:

The way that was chosen to implement the state machine diagram was to **solely** focus on the states of the program. The program has 6 states (starting from index 0) and each state has a different purpose. As the simulation runs, the state change based on certain events. See figure **1**.

A diagram of a computer

Description automatically generated

***Figure 1:*** State machine diagram for oven.

The states each have different names that correspond to the event that is occurring at the time of program run-time.

# Timer toggle:

Certain concepts need to be understood before this feature can be implemented. The idea was put together by breaking down and analysing the power button and the timer. Once powered on, the system should change the time only in **state 1** (when it is simply powered on and not performing any actions).

The button was added first on the graphical canvas by using the visualization toolbox and dragging a “push switch LED”. We then gave this a name by using the oven process object and assigning it a chosen name. In our case “i\_ChangeTimer\_Sw”. *Note: The before mentioned variable was defined in the “virtual inputs” section of the codebase with type BOOL.*

Finally, the state of the variable is checked and the timer variable that assigns the time component will be changed based on it. the conclusion behind this is because the timer is called a function block which can accept multiple variables. The time variable is one of them, meaning we didn’t need to make a separate function for it but simply pass in a different variable to it.

# Remarks:

*This section is for a small remarks that could be given.*

A remark that can be given is to add an additional feature in between state 1 and state 2. Two events need to be triggered to go from one state to the other. The remarks stems from the thought if an issue were to occur in between state 1 and state 2. This could provide a better idea of where the problem/bugs is located if one where to occur.

The initial Approach that was taken to implement the button was making use of the variables of type “fb{variable name}”. When this solution didn’t work, the structured text presentation came in quite useful. Specifically the slide that explains “Online Monitoring”. From there I realized that I was not using the correct variable type. When this was changed, the program ran as I intended.

The program can be optimized I feel using better coding style and making it error proof. For now this is the solution that was chosen because I don’t have any knowledge on this language yet.

# Bibliography:

**[1] –** Fontys PowerPoint. (n.d.). *SI -4 M&M Industrial Automation*