Prelab Section:

In this lab, we will use the PLC to control a run lamp with two different methods: using timers in the first experiment and counters in the second. The PLC system has already been set up and connected to the necessary components: the start and stop buttons as inputs, and the run lamp as an output. The SoMachine software was used to configure the PLC and program the logic.

In the first experiment, the timer block will be used to control the blinking of the lamp based on time delays. In the second experiment, the counter blocks (Count_UP and Count_D) will control the lamp based on counting sequences. The start button will initiate the process, and the stop button will reset the counter.

The objective is to learn how to use timers and counters in PLC programming to control outputs in real-world systems.

Fıgure : Photo from labrotory

Procedure Section:

Task *3*

What was implemented ?

In this lab, a simple automation circuit was designed to control a run lamp using a start button, a timer, and a "blink" block. When the start button connected to input terminal I0 is pressed, the run lamp connected to output terminal Q0 turns on. The objective of this experiment was to learn how to configure a timer to set a minimum trigger duration, use a blink block to set a blinking period, and observe the circuit operation in SoMachine software.

The system is designed such that the start button must be pressed for at least 1500 ms for the run lamp to turn on. TIMELOW and TIMEHIGH values were configured to set the on/off duration of the lamp's blinking. When the start button is pressed, the lamp begins blinking, and it stops blinking when the stop button is pressed. These operations were monitored and verified in SoMachine software.

How was it implemented ?

To implement this experiment, initial circuit connections were made by assigning input terminal I0 to the start button and output terminal Q0 to the run lamp. The second terminals of the button and the lamp were also connected to complete the circuit. In the SoMachine software, components like the "blink" block and timer were added. The timer played a crucial role in regulating the time duration for which the lamp would turn on. By setting the timer to activate the run lamp only after the start button was pressed for at least 1500 milliseconds, we ensured that a deliberate action (pressing the button for a specific duration) was required to trigger the lamp. This prevents the lamp from turning on accidentally and ensures precise control over the system's behavior.

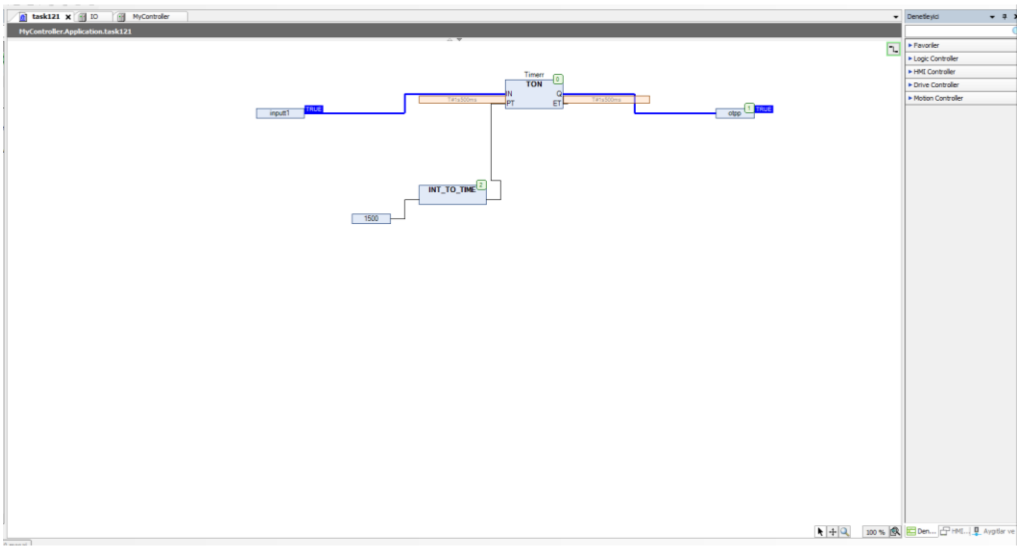

Figure 1: Photo from so machine application for task 3-1

The importance of using a timer lies in its ability to create precise delays, making it possible to manage the timing of events in the automation process. In this experiment, the timer allowed the "blink" block to control the on/off intervals of the lamp. By adjusting the TIMELOW and TIMEHIGH parameters, the blinking pattern could be modified, ensuring the lamp blinked with the desired frequency. Without a timer, it would be impossible to have such controlled delays, making it difficult to implement timed actions like blinking. The experiment was validated by observing the input and output states in SoMachine, confirming that the timer functioned as expected and enabled the lamp to operate at the correct timing intervals.

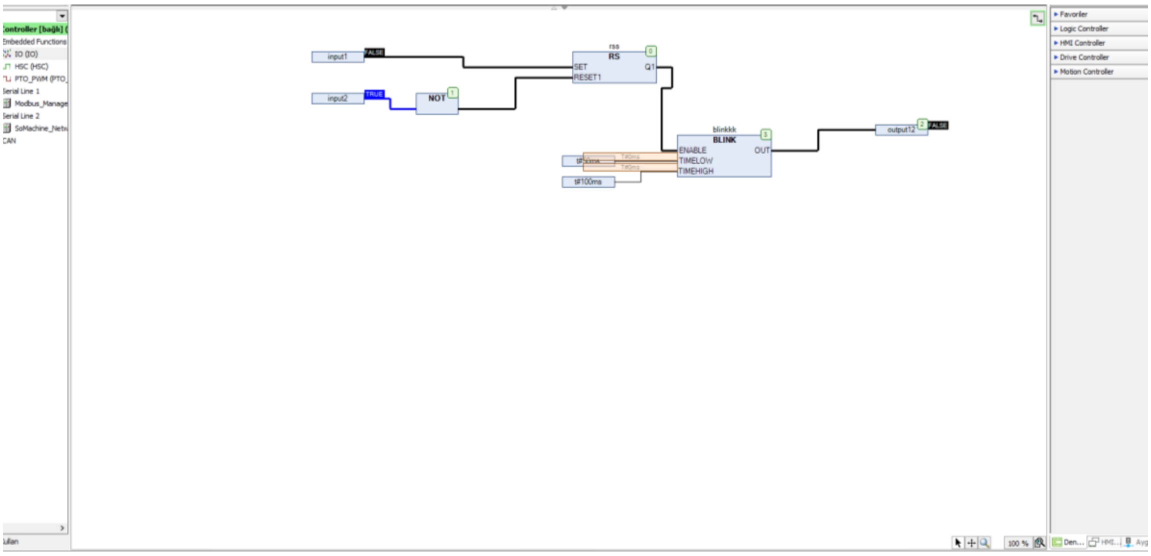

Figure 2: Photo from so machine application for task 3-2

Task *4*

What was implemented ?

In this experiment, a counter block was used to control the operation of a run lamp. The start button (myInput1) and stop button (reset) were used to initiate and reset the counting process, respectively. The run lamp (myOutput1) was used to indicate the output state. The objective of the experiment was to count up using the Count_UP block, and once the preset value (PV) was reached, the output would be set to TRUE, which would turn the lamp on. However, in order to light the lamp, a more complex process was used: the Count_UP block was employed to count five times, and once the Count_UP block had counted up five times, the Count_D block was used to count down. After the countdown, the run lamp would turn green, indicating the successful completion of the counting process.
The counter block played a critical role in managing the counting process and ensuring that the lamp would only light up after a specific number of counts had been reached. The use of both the Count_UP and Count_D blocks allowed for precise control over the timing and sequence, ensuring that the lamp would light up only after the correct sequence of counting and counting down was completed.

How was it implemented ?

The implementation of this experiment began by establishing the necessary connections in the SoMachine software. First, the start button was connected to the input terminal (myInput1), and the second terminal of the button was correctly wired to complete the circuit. Similarly, the stop button was connected to the reset terminal to reset the counter when needed. The run lamp was connected to the output terminal (myOutput1) so that its status could be observed during the experiment.

Once the basic connections were in place, the Count_UP block was added to the system. This block was used to initiate the counting process when the start button was pressed. The Count_UP block was configured with a preset value (PV), which determined the number of counts before the output was set to TRUE. As the start button was pressed, the counter counted up, and once the preset value was reached, the output turned TRUE, which would trigger the lamp to turn on.

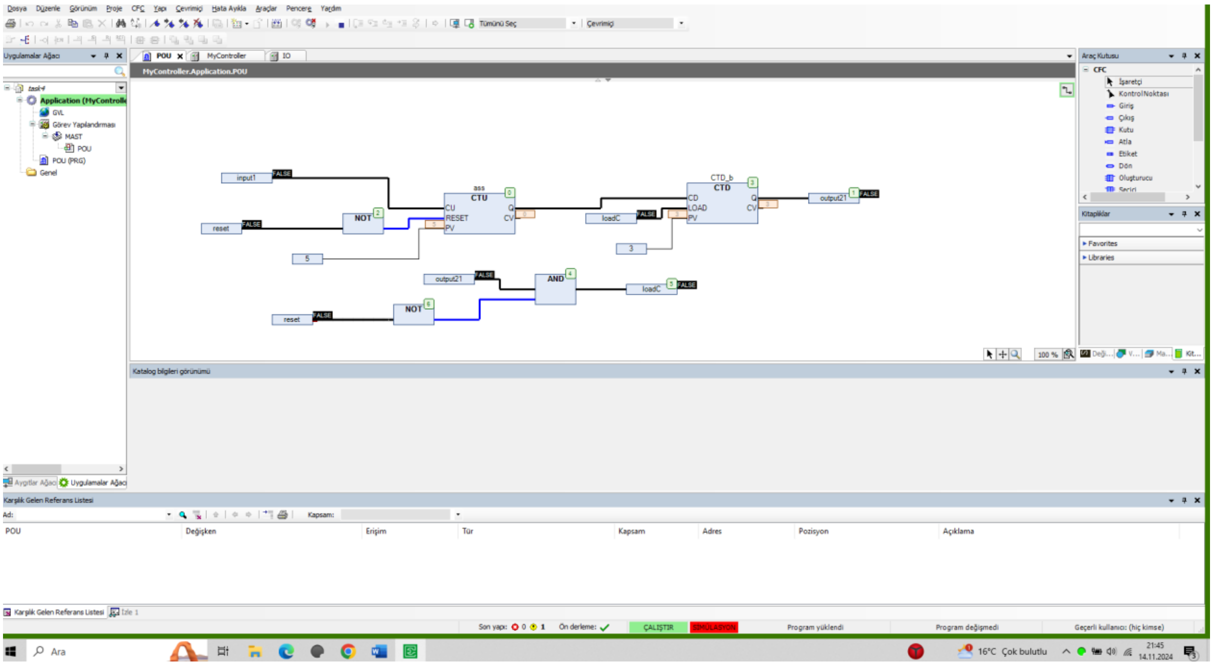

Figure 3: Photo from so machine application for task 4-1

However, simply counting up was not enough to turn on the lamp. The Count_D block was introduced to initiate the countdown process, and the lamp would only turn on after this countdown. The Count_UP block was configured to count five times, and once this count was completed, the Count_D block started counting down. This process was essential for making sure the lamp would only light up after the counter had completed the five counts. After the countdown was finished, the lamp turned green as expected.

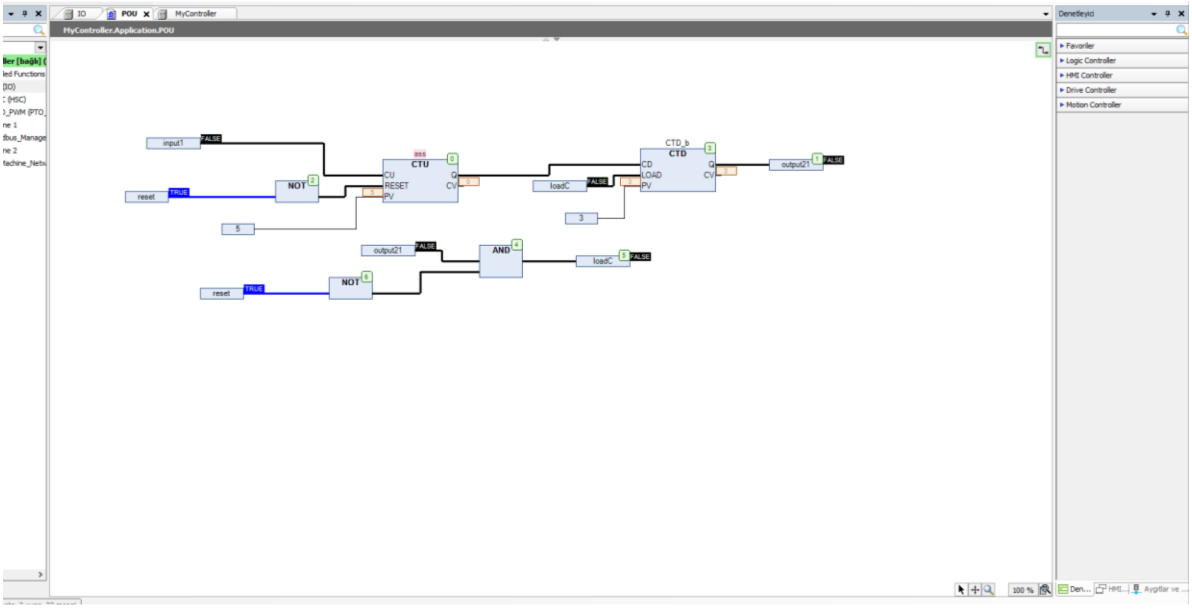

Figure 4: Photo from so machine application for task 4-1

To ensure the system worked correctly, the input and output states were monitored within the SoMachine software. When the start button was pressed, the counter began counting up. Once the Count_UP block reached the preset value, the output was set to TRUE, and the lamp began to light up. The Count_D block then began counting down, and after completing the countdown, the lamp turned green. This sequence confirmed that the counter blocks were functioning correctly, and the lamp was operating as expected.

Result

Both experiments successfully demonstrated the application of different control blocks to regulate the operation of a run lamp based on input conditions. In the first experiment, the use of a timer block allowed the lamp to blink when the start button was pressed and stopped when the stop button was pressed. The timer effectively managed the on and off times, creating a blinking pattern based on the adjusted TIMELOW and TIMEHIGH values. This experiment validated the importance of timers in controlling delays and the behavior of output devices in automated systems.

In the second experiment, the counter blocks (Count_UP and Count_D) were used to control the lamp based on a specific count sequence. The lamp turned on after five counts were completed, which was achieved by counting up with the Count_UP block and then using the Count_D block to count down. The setup also demonstrated how to manage timed actions with counters, where the output (lamp) only activated once a certain preset count was reached. The successful completion of both experiments confirmed the functionality of timers and counters in controlling outputs based on timed or counted conditions, which are essential components in automation systems.

Both experiments emphasized the significance of using control blocks to manage the timing and sequence of events in industrial automation, providing a deeper understanding of how to apply these blocks to control real-world devices like lamps.