

**JSC “Kazakh British Technical University”**

**Faculty of Information Technology**

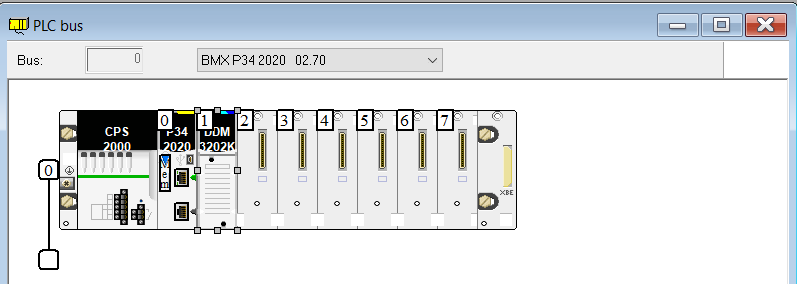
**Microprocessor Control Systems**

**Laboratory Work #1**

**Prepared by: Maratuly Temirbolat**

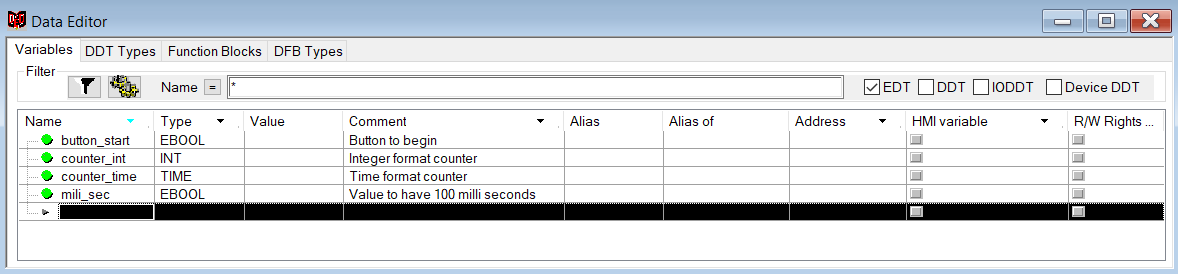
**Almaty 2021**

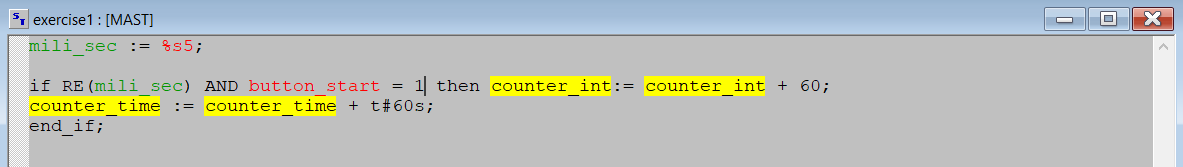
The whole laboratory work was done using the BMX P34 2020 Program Logical Controller. As the first module the DDM with 24 pins. Discrete Digital Module.

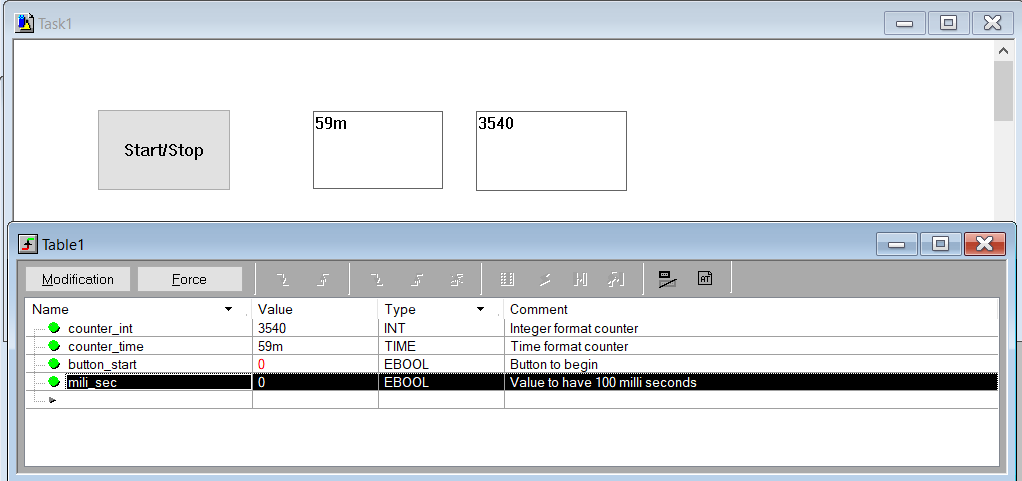


**Task #1.4**

Create a program that has two counters. One counter is of integer type, the second is of time format. It is necessary to program the increment of counters to 60 for an integer and 60 seconds for a counter working with the time format. The increment of the counters must take place in a time interval of 100 ms. The counters will be started after pressing the "START" button, the counters will be stopped after pressing the "STOP" button. Render the program.





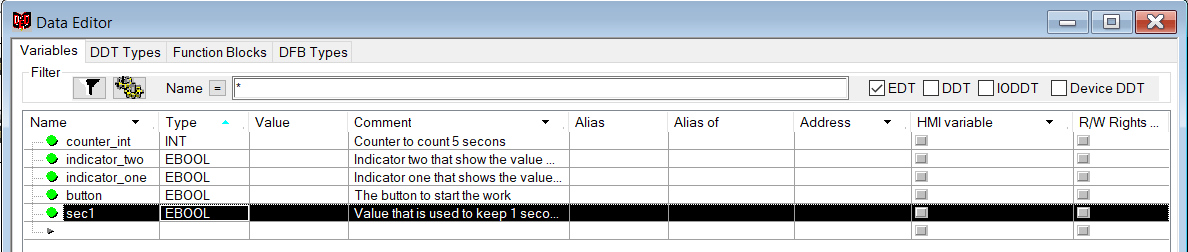


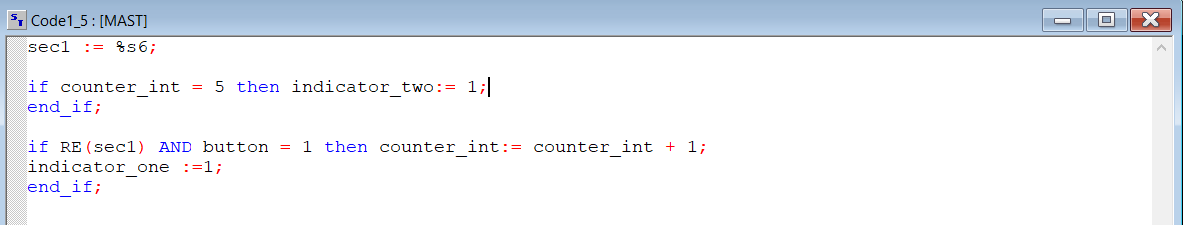
**Explanation:**

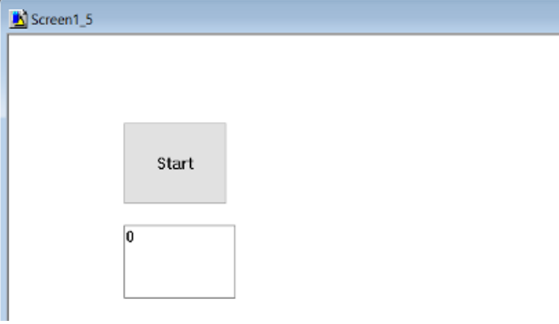
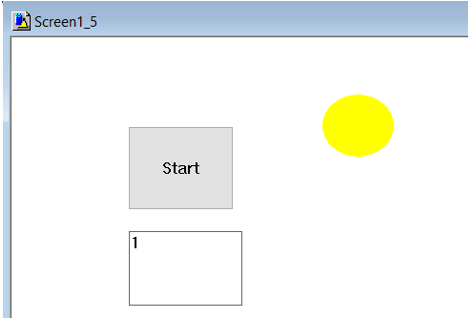
For the first exercise we draw and connected 3 items (1 Button that is responsible for the Beginning and End of the work, 2 counters (One Integer, One for Time)). Generally, there were created 4 variables (1,1,1 with button, counter\_int, counter\_time names respectively and with EBOOL, INT and TIME types respectively. As for the code part, there was written only one part where we used Logical comparison in order to use RE function and check whether the button is pressed. As a result, we increase the counter\_int by 60 and counter\_time by 1 minute or 60 seconds repeatedly after 1 second.

**Task # 1.5**

Two indicators of different colors are given. After pressing the "START" button, the first indicator lights up, after 5 seconds the second indicator turns on. Program this sequence with a counter. Render on a live screen.



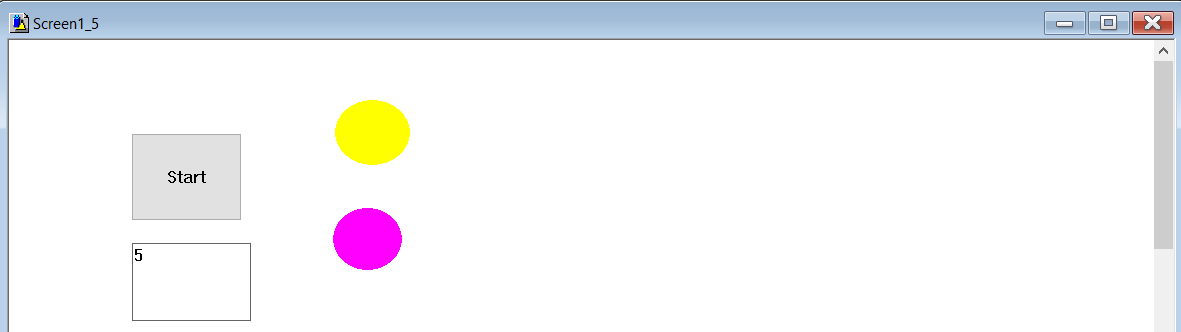


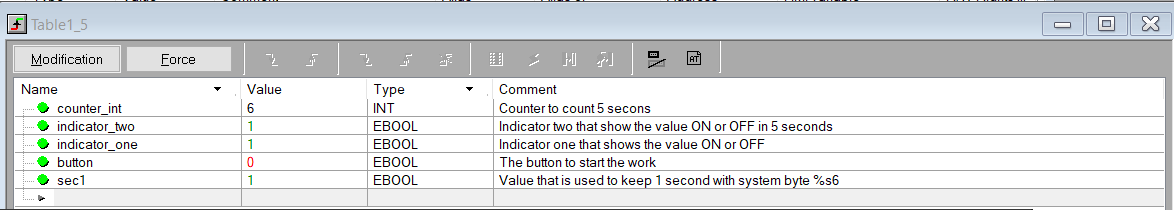


🡪

🡪

🡪



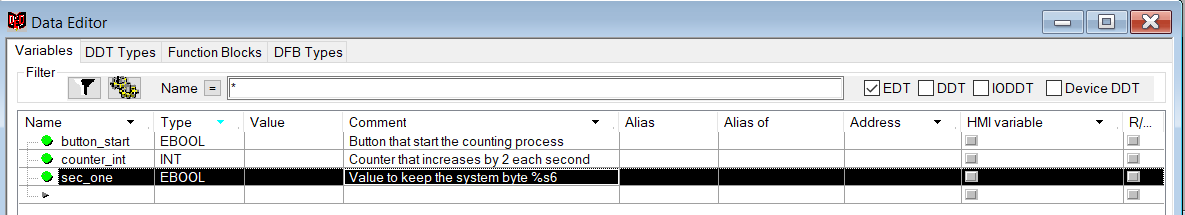


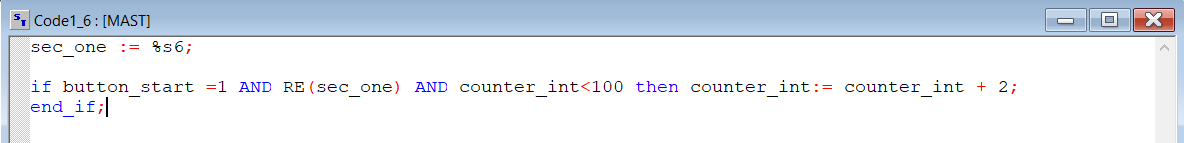
**Explanation:**

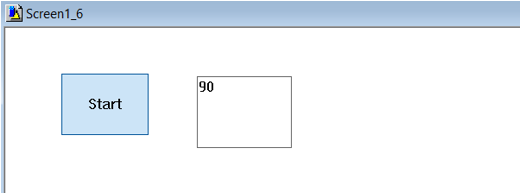
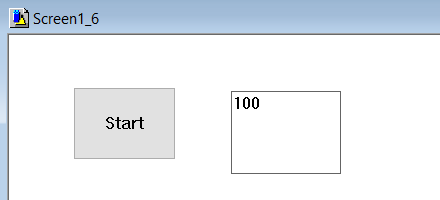
For the second exercise there were used 1 button to start working the counter, 1 counter to see the result and 2 round lights (first was yellow and second pink). The principle of the code is to check when the 5-th second is reached for which the counter is used because I increment the value each second to turn on the indicator number two. The first indicator works exactly after the pressing Start on the Button. For one second there was used system bit %s6 and written to the sec1 variable. It can be clearly seen that for the counter\_int was used INT type, for both indicators, button and sec1 EBOOL type.

**Task #1.6**

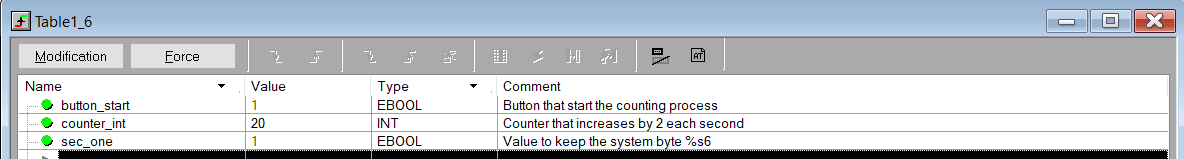
A counter is given, the value of which is incremented by 2 with an interval of 1 second, as soon as the value 100 is reached, the counter stops. The counter is started using the "START" button. Make a visualization of the program's work.







🡪

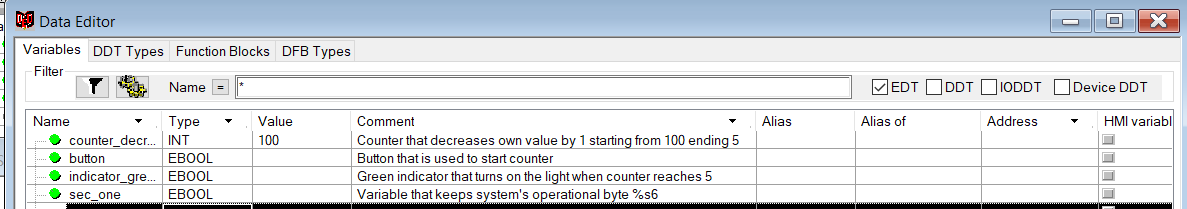


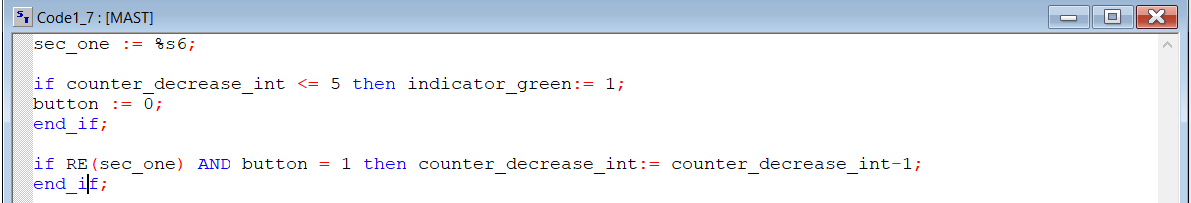
**Explanation:**

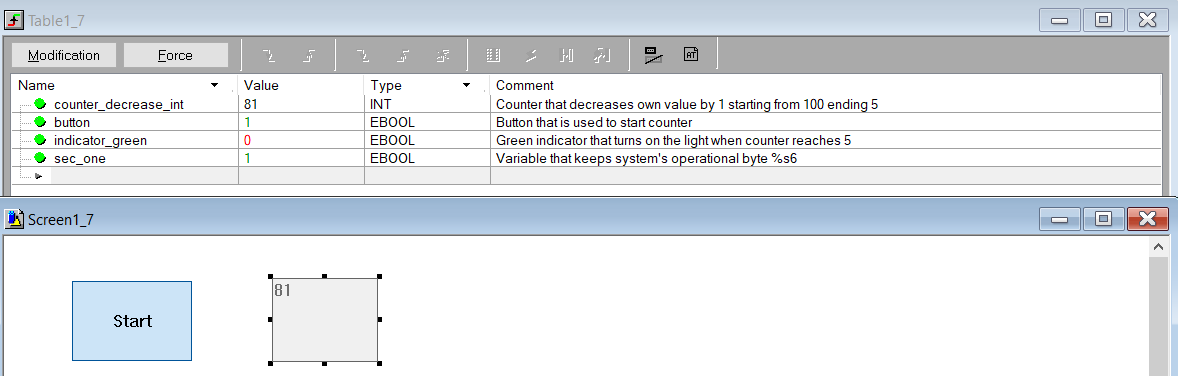
The third exercise has the button Start and display to see the counter’s value. The code contains only one if part where increase the value of counter by 2 if the button is pressed to work, time is going each second and counter’s number is less than 100 because then we can not rise it. As for variables: button and sec\_one have EBOOL and counter\_int INT data types.

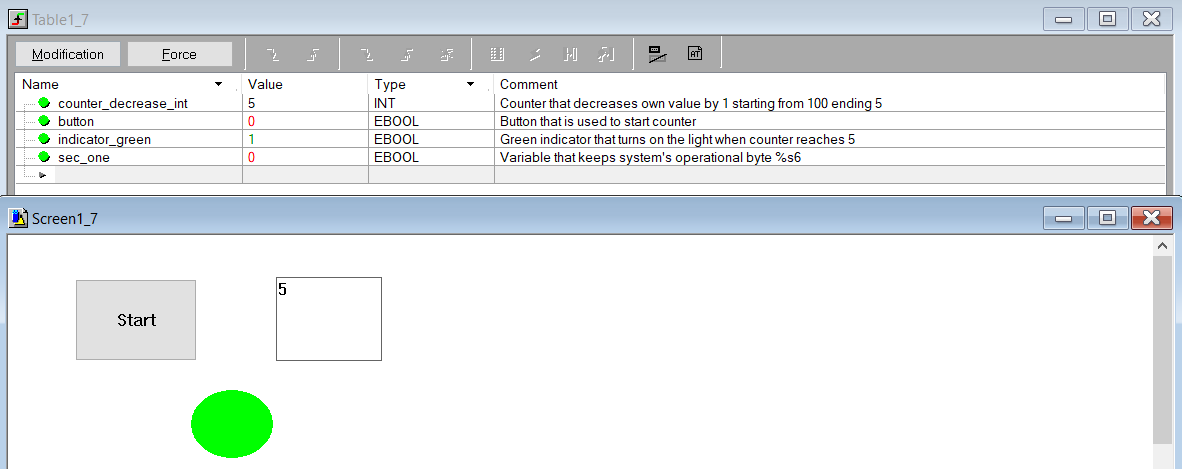
**Task #1.7**

The default value of the variable is 100. After pressing the START button, the counter starts counting down, as soon as the value of the variable reaches 5, the counter stops and the green indicator turns on. Make a visualization of the program's work.









**Explanation:**

The fourth final part of the work consists of 1 button, 1 display to see the value of counter and round green indicator that is turned on when counter achieves 5 seconds. The code illustrates that firstly we change the value of indicator from 0 to 1 and turn off the button if the value of counter is less or equal 5. The sign less or equal is used because we subtraction can be not using 1 or 5, instead of them can be used 3 and so on. Since of that this sign was used. The second logical comparison was used in order to start and continue decreasing the value of the counter by 1. For the sec\_one variable we assign 1 second bit to make an operation each second. As for the variables: counter\_deacrease\_int has INT data types, but indicator\_green, button, sec\_one have data types.

**Control questions:**

1. Describe the basic constructs of the ST language?
2. What is the RE command for?
3. What is a system bit?
4. Describe the purpose of the% S6 system bit?
5. What kind of counters are there?
6. How is time set in Unity Pro?
7. What are the SET and RESET commands for?
8. What is the function of the INC and DEC commands?

**Answers:**

1. ST editor is used for programming in structured text language. The ST programming language works with "Expressions". Expressions are composed of operators and operands that return a value after execution. Operators are symbols for the operations that are performed. Operands are variables, characters, inputs and outputs of functions and function blocks, etc. The code is formed in the form of instructions that are used to assign values returned from expressions. In the Unity Pro software environment, the working window of the ST editor is limited to 300 characters, and the section size is not limited and depends only on the memory size of the programmable logic controller.
2. The RE function is used to set the leading edge. In this case, we must work with the ebool variable type to set the leading edge. The RE function defines a 0 to 1 transition (Rising edge) of the associated bit.
3. A bit is a standard unit of measure that is used to measure information or data in computational and digital communications.
4. In the Unity Pro software environment, to create a counter, you can use system bits. Consider system bit% S6, which changes the time interval by 1 second
5. There are two types of counters in Unity Pro: signed (data type INT), unsigned (data type UDINT). A distinctive feature of these formats is that a signed INT counter can take values from 0 to 32768, and an unsigned counter can take values from minus infinity to plus infinity.
6. The time format counts down minutes and seconds. The time format in Unity Pro is set as follows: **t#1s**
7. The **SET** statement sets the variable to 1, the **RESET** statement sets the variable to 0.
8. The **INC** increases the value of the variable by 1 and keep this resulted number inside the value that was used, **DEC** does the same, but instead of increasing it decreases the figure by 1.

**Conclusion**

During the laboratory work we faced the Unity Pro program for the first time to start programming our controller. The ST language was used to code the parts of the system and connect to each other that is quite similar to C++, C# and other program languages. While the programming we revise such functions as INC, DECR and studied new (RE,SET,RESET). Also, there was introduced such variable which has TIME data type for which we used the conversion t#1S in order to work with it and sow the new data type as EBOOL that has about the same properties. The different types of counters were used and explained the difference between them. There were some problems with code parts, but after second attempt to do the work the problem disappeared and there were problems with the RE function because it did not work sometimes and instead of that I used just comparison conditions.