**TECHNICAL UNIVERSITY OF MOLDOVA**

**FACULTY OF COMPUTERS, INFORMATICS AND MICROELECTRONICS**

**DEPARTMENT OF SOFTWARE ENGINEERING AND AUTOMATICS**

**Laboratory work #6**

**Finite State Machine.**

**Performed the work:**

**Basso Nicolae, FAF-191**

**Checked by:**

**Moraru Dumitru**

**2022**

**1. The task of the laboratory work**

Sa se realizeze o aplicatie ce va implementa Automatele finite dupa cum urmeaza:

1. Proiectare Automat Finit aplicatie Button-Led.
2. Proiectare Automat Finit aplicatie Semafor

Recomandare:

a se utiliza interfata seriala pentru rapoarte de functionare a automatelor

Reutilizati la maxim solutiile prezentate in laboratoarele precedente

revizuiti resursele predate la curs

Pontaj

nota 5 - simpla aplicatie de comunicare

+1.0 - pentru implementare modulara a proiectului

+1.0 - pentru prezentarea si explicarea diagramelor Automatelor finite Button-Led

+1.0 - pentru prezentarea si explicarea tabelelor de tranzitii Button-Led

+1.0 - pentru prezentarea si explicarea diagramelor Automatelor finite Semafor

+1.0 - pentru prezentarea si explicarea tabelelor de tranzitii Semafor

+1.0 - pentru demonstrarea probelor de implementare fizica

penalitati

-1 - penalizare pentru NE utilizarea STDIO

-1 - penalizare pentru fiecare saptamana intarziere de la deadline

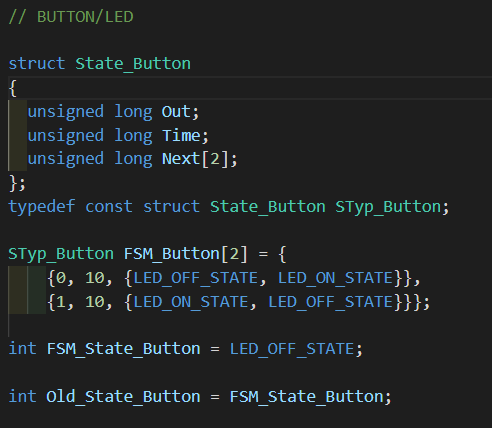
-1 - penalizare pentru nerespectare format raport

**2. The progress of the work**

**2.1 Description of the main functions used to perform the tasks**

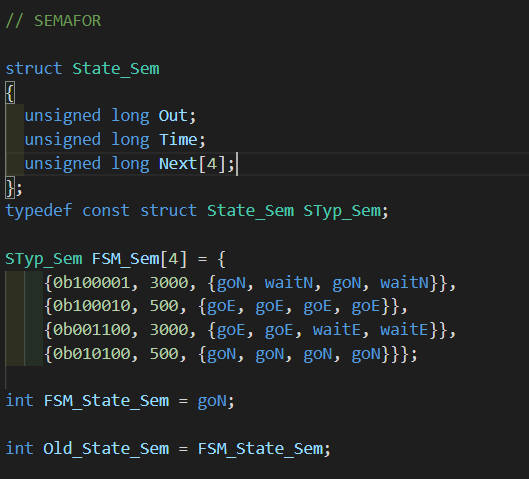
The problem was to create two Finite State Machines that could work with arduino, get input from the outer part of the system and affect on the outer part of the system:

1. Button / Led Finite State Machine, that has output for current state, time delay for current state and possible next states; in addition, there is memoryzation of the previous state:



**Figure 1** The button / led FSM code for the task.

1. The same way was implemented Finite State Machine for semaphore. As it is given in the task, in fact there are two semaphores – both are located perpendicular to each other (NORTH and EAST in Proteus Project). There are more states, than in the previous example:

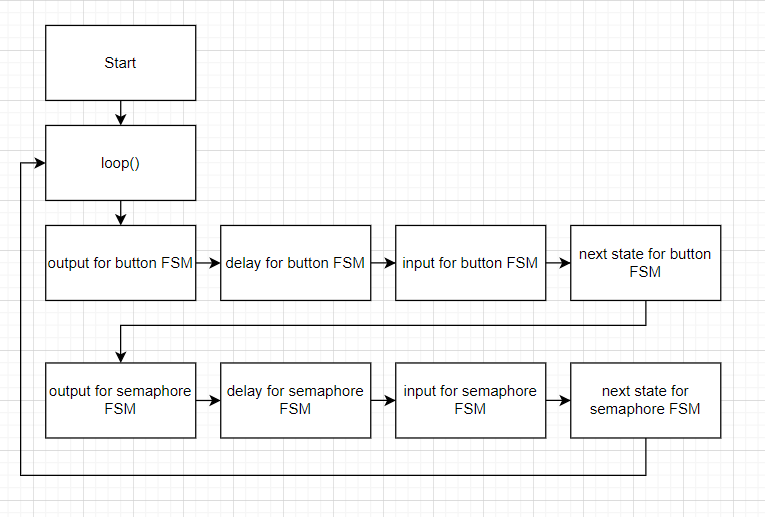


**Figure 2** The semaphore FSM code for the task.

**2.2 The programs’ diagram block**

1) Flowchart diagram for the sequential type of software:

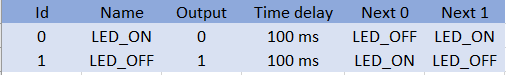
During this part of laboratory work, all the code is being executed consequently, all the main code is located in several main function and classes that process data. Of course, these functions contain operations and other functions invocations. Afrer all the data was received, it prints it.



**Figure 3** Flowchart diagram for the main functions.

2) Button / LED Finite State Machine Table:

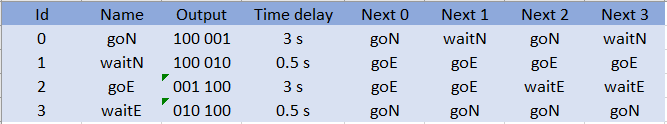
On the figure below is presented Table of Finite State Machine of Button / LED part, here are presented id of the state, its name, output, time delay, and next possible states from the current state:



**Figure 4** Table for Button / LED state machine

3) Semaphore Finite State Machine Table:

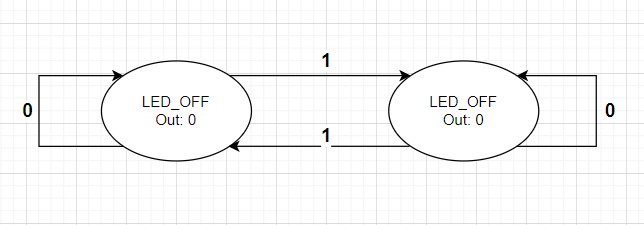
On the figure below is presented Table of Finite State Machine of Semaphore part, here are presented id of the state, its name, output, time delay, and next possible states from the current state:



**Figure 5** Table for Semaphore state machine

4) Button / LED Finite State Machine Diagram:

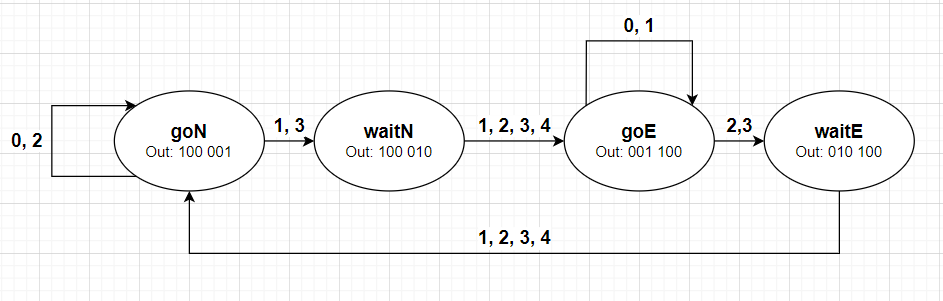
On the figure below is presented Diagram of Finite State Machine of Button / LED part, here are presented all possible states and transitions between them:



**Figure 6** Diagram for Button / LED state machine

5) Semaphore Finite State Machine Diagram:

On the figure below is presented Diagram of Finite State Machine of Semaphore part, here are presented all possible states and transitions between them:

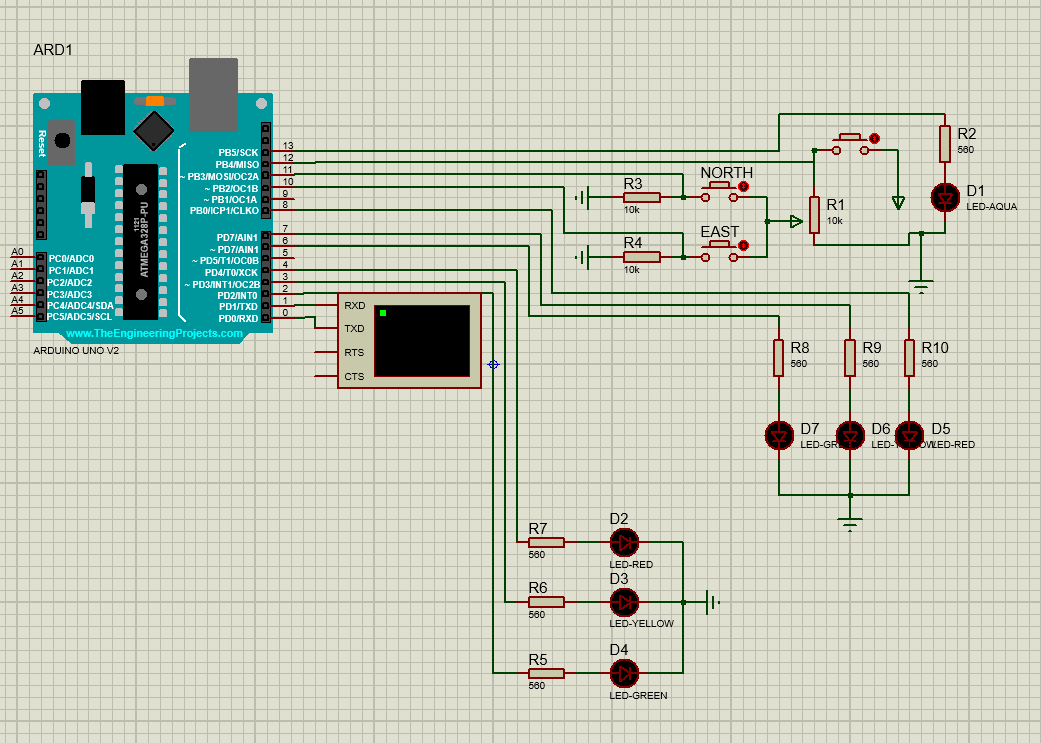


**Figure 7** Diagram for Semaphore state machine

**2.3 Simulated electrical schematic**

1) Schematic of the solution for the sequential type of software:

MCU-based application that controls the drive devices with commands received from the buttons and reporting to the Serial terminal.



**Figure 8** Simulated schematic for the solution.

**Conclusions**

During this laboratory work, we implemented an MCU-based application that controls the devices with commands received from buttons and some data printed using Serial terminal. The two finite state machines are controlling the whole system by using states and transitions between them. The first Finite State Machine is easier to manage, build and implement, the second one is way harder. The first Finite State Machine is similar to the application that was developed for the first laboratory work, but using Finite State Machine principle, where the only value is transmitted to the led, here is changed to the state that can be modified in real time using buttons or other control devices. The second Finite State Machine is simulating the semaphore behavior, where the first semaphore is turning green, the other one first of all goes to the yellow light state, and then to the red light, and vice versa.

**Bibliography**

1. ARDUINO: *Arduino Language Reference*. Arduino official site, ©2022 [quote 28.04.2022]. Access link: <https://www.arduino.cc/reference/en/>

  Что вершит судьбу человечества в этом мире? Некое незримое существо или закон, подобно Длани Господней парящей над миром? По крайне мере истинно то, что человек не властен даже над своей волей.