

# American International University-Bangladesh

## Dept. of Computer Science and Engineering

Course Name: MICROPROCESSOR AND EMBEDDED SYSTEMS

Section: M Session: Spring 21-22 Group No: 03

Course Instructor: Dr. Ferdous Jahan Shaun

Submission Date: 17-02-2022

# Title: Familiarization with Visual Designer for Arduino™ AVR and Raspberry Pi and implementation of a traffic control system using Drag - Drop - Play.

**Group Members**

|  |  |
| --- | --- |
| **Name** | **ID** |
| Mohammad Sakib Islam Mazumder | 19-39290-1 |
| Munira Zebin | 19-41014-2 |
| Samir Faisal | 19-41037-2 |
| Nasim Reza Hridoy | 19-41045-2 |
| Shafiul Islam Mazumder | 19-41370-3 |
| Arpita Saha | 19-41363-3 |
| Puravi Debnath Nitu | 19-41371-3 |
| S. M. Rizwanul Hassan | 19-41386-3 |

Contents

[Title: Familiarization with Visual Designer for Arduino™ AVR and Raspberry Pi and implementation of a traffic control system using Drag - Drop - Play. 1](#_Toc95774539)

[1. Abstract 3](#_Toc95774540)

[2. Objectives 3](#_Toc95774541)

[3. Results 3](#_Toc95774542)

[1) Simulation Environment 3](#_Toc95774543)

[2) Simulation Result 6](#_Toc95774545)

[3) Discussion 8](#_Toc95774548)

[4. Question for Report writing 9](#_Toc95774549)

[5. Conclusion 10](#_Toc95774550)

[6. References 11](#_Toc95774551)

[Figure 1: Proteus 4 Phase-1 3](#_Toc95774599)

[Figure 2: Proteus 4 Phase-2 3](#_Toc95774600)

[Figure 3: Proteus 4 Phase-3 4](#_Toc95774601)

[Figure 4: Proteus 4 Phase-4 4](#_Toc95774602)

[Figure 5: Proteus 4 Phase-5(Raspberry pi) 4](#_Toc95774603)

[Figure 6: Proteus 4 Phase-6 4](#_Toc95774604)

[Figure 7: Proteus 4 Phase-7 5](#_Toc95774605)

[Figure 8: Proteus 4 Phase-8(Raspberry pi) 5](#_Toc95774606)

[Figure 9: Proteus 4 Phase-9(Arduino uno) 5](#_Toc95774607)

[Figure 10: Proteus 4 Phase-10(Arduino uno) 5](#_Toc95774608)

[Figure 11: Traffic Control system circuit design using Raspberry pi 6](#_Toc95774609)

[Figure 12: Traffic control system Flow chart 7](#_Toc95774610)

[Figure 13: Traffic Control system circuit design using Arduino Uno 8](#_Toc95774611)

[Figure 14: two-way traffic control system Arduino circuit simulation 9](#_Toc95774612)

[Figure 15: two-way traffic control system flow chart 10](#_Toc95774613)

# Abstract

The main goal of this experiment is to gain knowledge about how to build “Traffic Control system” in Visual Designer and Schematic capture, as well as to become familiar with the Proteus 8 Professional software. This is an effective software basically designed for the developers and student to develop, design and test circuit boards quite easily. In this lab report, firstly we learned how to design flow chart and how to connect it to the Arduino uno board and Raspberry Pi board. The main difference between this two-type board is Arduino is microcontroller board on the other hand Raspberry Pi is a microprocessor-based minicomputer. Which mean The Microcontroller on the Arduino board contains the CPU, RAM and ROM and all the additional hardware on the other hand Raspberry Pi has all features of a minicomputer with a processor, memory, graphics driver, storage etc. After built the flowchart, circuit performed successfully. And the traffic control works properly.

# Objectives

1. Learning to make embedded system using Drag - Drop – Play method.
2. Implementation of a traffic control system using Drag - Drop – Play method.

# Results

## Simulation Environment

## Proteus 4 Software

|  |  |
| --- | --- |
| Figure : Proteus 4 Phase-1 | Figure : Proteus 4 Phase-2 |

* Figure 1: Open Proteus software. Then Click New Project.
* Figure 2: Write the project name and then click next.

|  |  |
| --- | --- |
| Figure : Proteus 4 Phase-3 | Figure : Proteus 4 Phase-4 |

* Figure 3: Select DEFAULT and then click next.
* Figure 4: Select First Option and then click next.

|  |  |
| --- | --- |
| Figure : Proteus 4 Phase-5(Raspberry pi) | Figure : Proteus 4 Phase-6 |

* Figure 5: Select create Firmware Project and select Raspberry Pi the drop-down menu then clicks next.
* Figure 6: Click Finish.

|  |  |
| --- | --- |
| Figure : Proteus 4 Phase-7 | Figure : Proteus 4 Phase-8(Raspberry pi) |

* Figure 7: This is the dashboard of Visual designer in Proteus Software where we have to create flowchart.
* Figure 8: This is the dashboard of schematic capture in Proteus Software where we have to build the traffic control system circuit using Raspberry pi.

|  |  |
| --- | --- |
| Figure : Proteus 4 Phase-9(Arduino uno) | Figure : Proteus 4 Phase-10(Arduino uno) |

* Figure 9: Select create Firmware Project and select Arduino uno in the drop-down menu then click next.
* Figure 10: This is the dashboard of schematic capture in Proteus Software where we have to build the traffic control system circuit.

## Simulation Result

## Using Raspberry Pi

|  |
| --- |
| Figure : Traffic Control system circuit design using Raspberry pi |

* Figure 11: First go to file section>click Import Project Clip> Go to Folder "Raspberry Pi">select Raspberry LED Red Breakout> then click open > place it in the perfect position. Same process has to be done respectively for other two LED Breakout lights.

|  |
| --- |
| Figure : Traffic control system Flow chart |

* Figure 12: Initially all LED will turn off. After simulation first LED1 that’s mean RED LED will glow for 3 sec and simultaneously LED2 that’s mean Yellow LED will glow for half sec then Red LED will turn off. After 1 sec Yellow LED will off and simultaneously Green LED will on for 1 sec. After 1 sec Green Led will off. This process will continue for 3 sec which should like the LED will blinking. Then Blinking will stop. Then the yellow led will glow for 1 sec, then the red, continuously follow this process, in the end, the simulation done properly.

## Using Arduino Uno

|  |
| --- |
| Figure : Traffic Control system circuit design using Arduino Uno |

* Figure 13: First go to file section>click Import Project Clip> Go to Folder "Raspberry Pi">select Raspberry LED Red Breakout> then click open > place it in the perfect position. Same process has to be done respectively for other two LED Breakout lights.

Also need to design the flow Chart for the Arduino Uno circuit. The flow Chart is also same as Figure 12 and description is also same.

## Discussion

Basically, traffic control system has been done by two ways. One way is Raspberry Pi and another way is Arduino Uno. In first way, we using Raspberry Pi for implement the simulation. We used inbuild breakout LED in this simulation. Then we design a flowchart which is connected to the circuit. We used the same process for Arduino Uno. Finally, the traffic control system works properly.

# Question for Report writing

|  |
| --- |
| Figure : two-way traffic control system Arduino circuit simulation |

* Figure 14: For completing this task, two road will consider which will intersecting. First row determines one way road. And second row determines another way road. First, we have to go File section. Then select “Click import project clip”. A prompt will be shown where One pair of each LED (RED, GREEN, YELLOW) have to select. Two group of light have to place according to figure 14.

|  |
| --- |
| Figure : two-way traffic control system flow chart |

* Figure 15: First we have to build a flow chart which contain all off LED. Then we have to create a loop where LED4(Green) and LED1(Red) will glow for 3 sec. Then LED1(Red) remain on but LED2(Yellow) will off for 5 sec. In the meantime, LED5(Yellow) will on and LED4(Green) will off for 5 sec. Then LED5(Yellow) and LED2(Yellow) both will be blinking frequently for 3 sec. After LED5(Yellow) and LED2(Yellow) will off on the other hand LED3(Green) and LED6(Green) will simultaneously on. This process will continue for 3 sec. Then the whole process will repeat from start to end for complete the simulation. In the last all LED5(Yellow) and LED2(Yellow) will off.

# Conclusion

In this simulation, the goal was to build a traffic system. This simulation has done by Two different board. One is open source of the microcontroller board like Arduino Uno and another one microprocessor-based minicomputer like Raspberry Pi board were implemented to build the system. Priority was given to having no errors or difficulties in understanding. Everything is done perfectly.

# References

1. <https://www.coursera.org/learn/arduino/lecture/ei4ni/1-10-first-glance-at-a-program>.
2. https://www.arduino.cc/.
3. Jeremy Blue; Exploring Arduino: Tools and Techniques for Engineering Wizardry
4. https://www.raspberrypi.org/