**Traffic Signal Management & Control System Based on Density of Vehicles and Emergency Vehicles**

**Submitted By:**

* Farjana Afroz , Roll: 2009032
* Md. Shafe-ul-Alam , Roll: 2009035

**Objectives:**

1. Real-Time Traffic Density Monitoring
2. Dynamic Signal Control
3. Emergency Vehicle Priority
4. Automated Signal Coordination

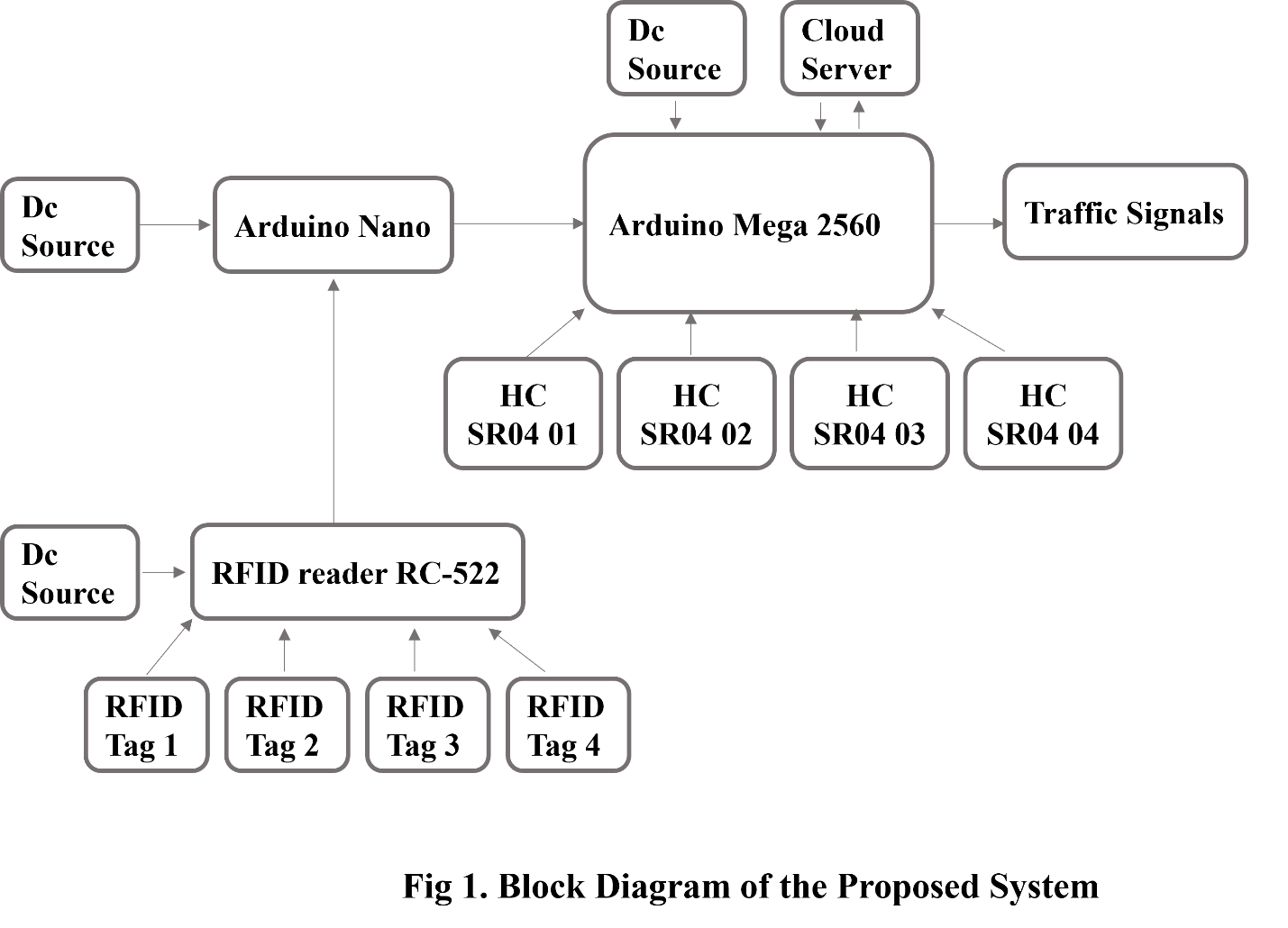
**Project Overview:**

This system is a smart solution that automatically manages and improves how traffic signals work. It intelligently adjusts traffic light timings by leveraging real-time data on vehicle density at intersections, helping to reduce congestion and keep traffic flowing smoothly. It also gives priority to emergency vehicles, such as ambulances or fire trucks, by changing the traffic lights to green as they approach, allowing them to pass through intersections quickly and safely.

**Features:**

* **Real-Time Traffic Density Monitoring**: The system employs various sensors to monitor the number of vehicles at an intersection in real time.
* **Dynamic Signal Timing Adjustment**: Based on the analyzed traffic density data, the system dynamically adjusts traffic light timings.
* **Reduce Traffic Congestion**: Minimize congestion and long wait times by dynamically managing traffic signals to respond to varying vehicle loads.
* **Emergency Override System**: Allows traffic authorities to manually override the automated system in case of emergencies or special events that require different traffic management.
* **Intersection Coordination**: Synchronizes adjacent traffic signals (green wave) to allow a smoother flow of vehicles through multiple intersections, especially during peak traffic hours.

**Block Diagram:**



**Equipment and Cost Estimation:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.** | **Item** | **Justification** | **Price (BDT)** |
| 1 | Breadboard | For testing and prototyping connections | 100 |
| 2 | Arduino mega 2560 | Main microcontroller for handling complex tasks | 1895 |
| 3 | Arduino nano | |  | | --- | | Secondary controller for smaller tasks | | 410 |
| 4 | HC SR04 Ultrasonic sensor | Detect vehicle density at intersections | 420 |
| 5 | RFID RC-522 module | To identify and prioritize emergency vehicles | 195 |
| 6 | RFID Tags | For emergency vehicle detection | 120 |
| 7 | 220 Ohm Resistors | Current limiting for LEDs and other components | 10 |
| 8 | Signal light Module PCB | For controlling the traffic signal lights | 120 |
| 9 | 12V DC power supply | Provides power to the system | 110 |
| 10 | 7 Segment Display | Displays real-time information on traffic signals | 250 |
| 11 | Red, Green, Orange, LED | Indicator lights for traffic signals | 20 |
| 12 | Dotted veroboard | |  | | --- | | For assembling the components | | 130 |
| 13 | 74HC595 | Shift register for controlling multiple LEDs | 120 |
| 14 | Male, Female Header | For making connections between components | 100 |
| Total (BDT) | | | 4000 |
| In words: Four Thousand Taka only | | | |

|  |  |
| --- | --- |
| Equipment | Quantity |
| Breadboard | -1 |
| Arduino mega 2560 | -1 |
| Arduino nano | -1 |
| HC SR04 Ultrasonic sensor | -8 |
| RFID RC-522 module | -1 |
| RFID Tags | -8 |
| 220 Ohm Resistors | -12 |
| Signal light Module PCB | -4 |
| 12V DC power supply | -1 |
| 7 Segment Display | -8 |
| Red, Green, Orange, LED | -15 |
| Dotted veroboard | -1 |
| 74HC595 | -4 |
| Female Header | -20 |
| Male Header | -20 |

|  |  |  |  |
| --- | --- | --- | --- |
| Roll | Course Title | Course Code | Obtained Grade |
| 2009032 | Solid State Electronics Laboratory | ECE1110 | A+ |
| Electronics Circuit Design Laboratory | ECE 2200 | A- |
| Microprocessors and Microcomputers | ECE 3103 |  |
| 2009035 | Solid State Electronics Laboratory | ECE1110 | A |
| Electronics Circuit Design Laboratory | ECE 2200 | A- |
| Microprocessors and Microcomputers | ECE 3103 |  |

|  |  |
| --- | --- |
| 1st week | Project initialization and planning |
| 2nd week | Project initialization and planning |
| 3rd week | Requirement analysis and gathering |
| 4th week | Requirement analysis and gathering |
| 5th week | Hardware and software selection and testing |
| 6th week | Hardware and software selection and testing |
| 7th week | System design and architechture |
| 8th week | System design and architechture |
| 9th week | Integration and testing |
| 10th week | Integration and testing |
| 11th week | Software developement |
| 12th week | User testing and feedback |
| 13th week | Final documentation and prepare for next development phase |