System Description

The traffic light controller system uses an ATmega32 microcontroller to control the traffic lights. The system is designed to switch between different LED states in response to different inputs from the push button and timer. When the push button is pressed, the pedestrian light will switch to green, indicating that it is safe for pedestrians to cross the road. The cars light will also switch to red, indicating that they should stop. After a fixed time, the pedestrian light will switch to red, indicating that pedestrians should no longer cross the road. The cars light will switch to green, indicating that they can now proceed. The yellow light will be used as a transition between green and red.

The system uses the INT0 pin to detect when the push button is pressed, and the port A and port B pins to control the LED lights. The microcontroller uses its internal timer to control the duration of each LED state and switch between them. The system is designed to be simple and reliable, with a focus on ease of use and maintainability.

System Design

The software design of the traffic light controller system is based on Layered architecture. The software components of the system are divided into three layers: Microcontroller Abstraction Layer (MCAL), Hardware Abstraction Layer (HAL), and the Application Layer.

- **Microcontroller Abstraction Layer (MCAL)**

The MCAL layer implements the low-level functions to interact with the microcontroller ATmega32. In this layer, the following components are implemented:

1. Digital Input Output (DIO) The DIO component is responsible for controlling the state of the input and output pins of the microcontroller.
2. Timer0 The Timer0 component is responsible for generating time delays for different tasks of the system.

3. External Interrupt The External Interrupt component is responsible for handling the interrupt request from the pedestrian push button.

- **Hardware Abstraction Layer (HAL)**

The HAL layer implements the high-level functions to interact with the hardware components of the system. In this layer, the following component is implemented:

1. LED Driver The LED Driver component is responsible for controlling the state of the traffic lights (cars and pedestrians).

- **Application layer (APP)**

the application layer is where the main function of the system resides. It makes use of the lower layers, such as the MCAL and HAL, to control the traffic lights based on the input from the push button. This layer is responsible for the overall functioning of the system, including the state transitions of the LEDs for cars and pedestrians.

- **Libraries (LIB)**

The software of the system also uses some libraries to provide standard data types, general purpose registers, error states and bitmasks:

1. Standard Data Types Library This library provides standard data types for the system, such as uint8, int16, etc.
2. General Purpose Registers Library This library provides access to the general-purpose registers of the microcontroller.
3. Error States Library This library provides the error codes for different error states that might occur during the system operation.
4. Bit level library, The Bit masking Library is an essential component of the software design for the traffic light controller system. It serves the purpose of providing a set of predefined bit masking values that can be used in the implementation of the drivers. The library provides a unified and standardized way of dealing with bit masking, making it easier to maintain and modify the code.

The above components and libraries are integrated together to form the complete software of the traffic light controller system.

Flow chart