Requirements:

* Hardware Requirements:

1. Atmega 32 Microcontroller.
2. One push button
3. Three LEDs for car traffic (Green – Yellow – red).
4. Three LEDs for pedestrian (Green-Yellow-red).

* Software Requirements:

1. In Normal mode:
2. Pedestrian red LED on.
3. Car’s traffic led on start from green for five second then yellow blink for five sec then red on for five and back to Yellow blink for five seconds and repeats the processing.
4. In pedestrian mode :
5. Change from normal mode to pedestrian mode when the pedestrian button is pressed.
6. If pressed when the cars' Red LED is on, the pedestrian's Green LED an If pressed when the cars' Red LED is on, the pedestrian's Green LED and the cars' Red LEDs will be on for five seconds, this means that pedestrians can cross the street while the pedestrian's Green LED is on.
7. If pressed when the cars' Green LED is on or the cars' Yellow LED is blinking, the pedestrian Red LED will be on then both Yellow LEDs start to blink for five seconds, then the cars' Red LED and pedestrian Green LEDs are on for five seconds, this means that pedestrian must wait until the Green LED is on.
8. At the end of the two states, the cars' Red LED will be off and both Yellow LEDs start blinking for 5 seconds and the pedestrian's Green LED is still on.
9. After the five seconds the pedestrian Green LED will be off and both the pedestrian Red LED and the cars' Green LED will be on.
10. Traffic lights signals are going to the normal mode again.

* System Layers:

1. Microcontroller
2. Application
3. ECUAL
4. MCAL
5. Utilities

APPLICATION LAYER

* System Driver :

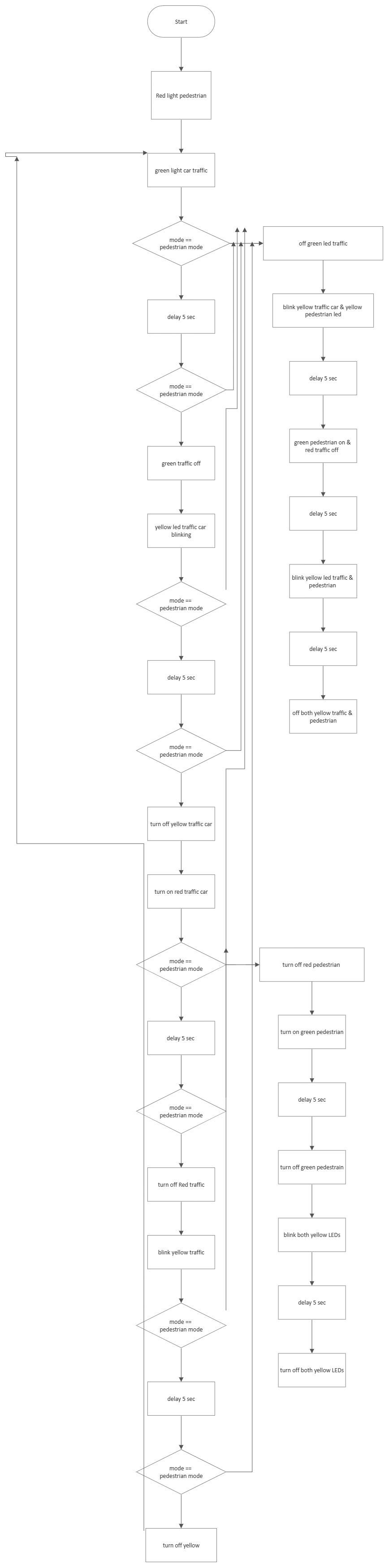
1. DIO
2. Timer
3. LED
4. Interrupt
5. Push Button

* **Data Type:**

1. **typedef unsigned char uint8;**
2. **typedef unsigned short uint16;**
3. **typedef unsigned int uint64;**
4. **typedef signed char int8;**
5. **typedef signed short int16;**
6. **typedef signed int int64;**

* **Flow Chart** :

**\*\*note 🡺 zoom in Flowchart**

****