Requirements:

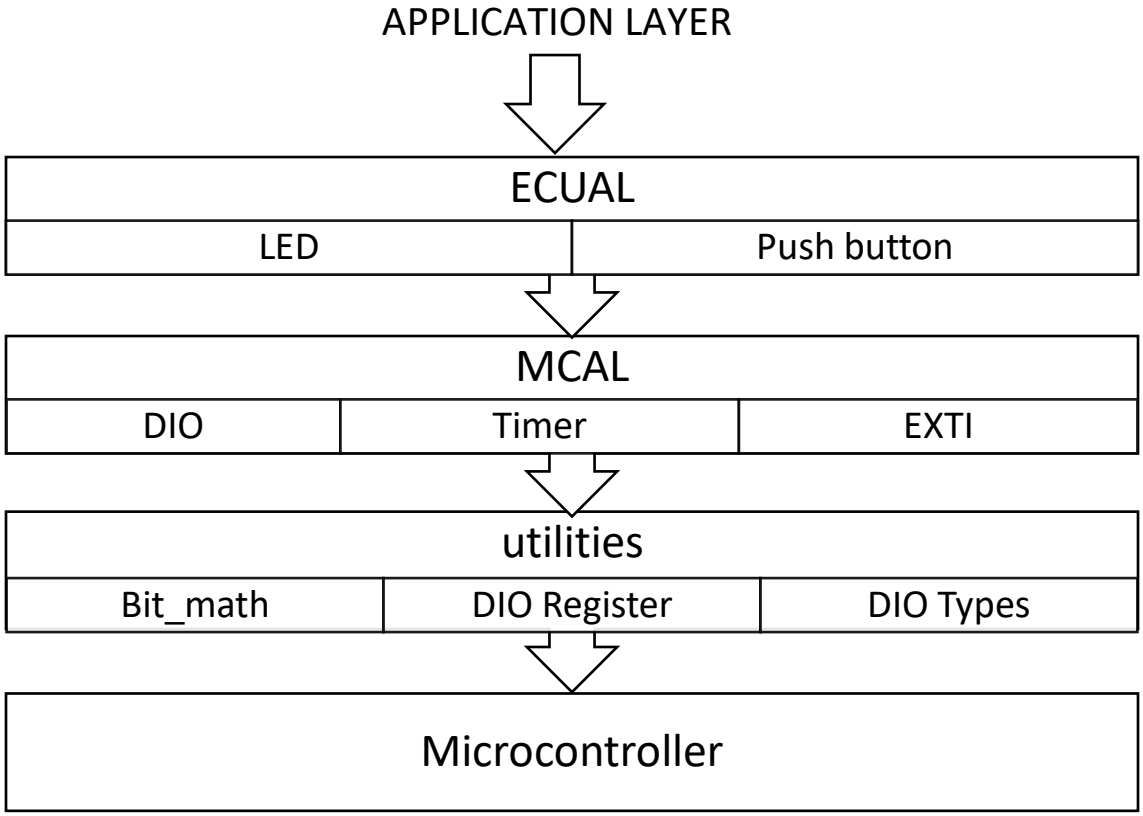
- Hardware Requirements:
  1- Atmega 32 Microcontroller.
  2- One push button
  3- Three LEDs for car traffic (Green – Yellow – red).
  4- Three LEDs for pedestrian (Green-Yellow-red).

- Software Requirements:
  1- In Normal mode:
     a. Pedestrian red LED on.
     b. 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.

  2- In pedestrian mode :
     a. Change from normal mode to pedestrian mode when the pedestrian button is pressed.
     b. 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.
     c. 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.

d. 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.
e. 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.
f. Traffic lights signals are going to the normal mode again.

- System Layers:
  1- Microcontroller
  2- Application
  3- ECUAL
  4- MCAL
  5- Utilities



- System Driver :
  1- DIO
  2- Timer
  3- LED
  4- Interrupt
  5- Push Button

DIO APIS

- void DIO_Set_PortDirection(DIO_PORT_ID PORT_ID ,DIO_Direction Direction);
- void DIO_Set_PortVal(DIO_PORT_ID PORT_ID , uint8 value);
- void DIO_Set_Pin_Direction(DIO_Pin_ID PIN_ID ,DIO_PORT_ID PORT_ID , DIO_Direction Direction) ;
- void DIO_Set_Pin_Val(DIO_Pin_ID PIN_ID ,DIO_PORT_ID PORT_ID , DIO_Value value);
- DIO_Value DIO_Get_Pin_Val(DIO_Pin_ID PIN_ID , DIO_PORT_ID PORT_ID);
- void DIO_toggle(uint8 pin_id, uint8 port_id);

Timer APIS

- void Timer0_init(ST_Timer_Config Prescaler);
- void timer0_start(uint8 value);
- void timer0_Get(uint8 * value);
- void Timer_stop();
- void Timer_delay(float msecend);

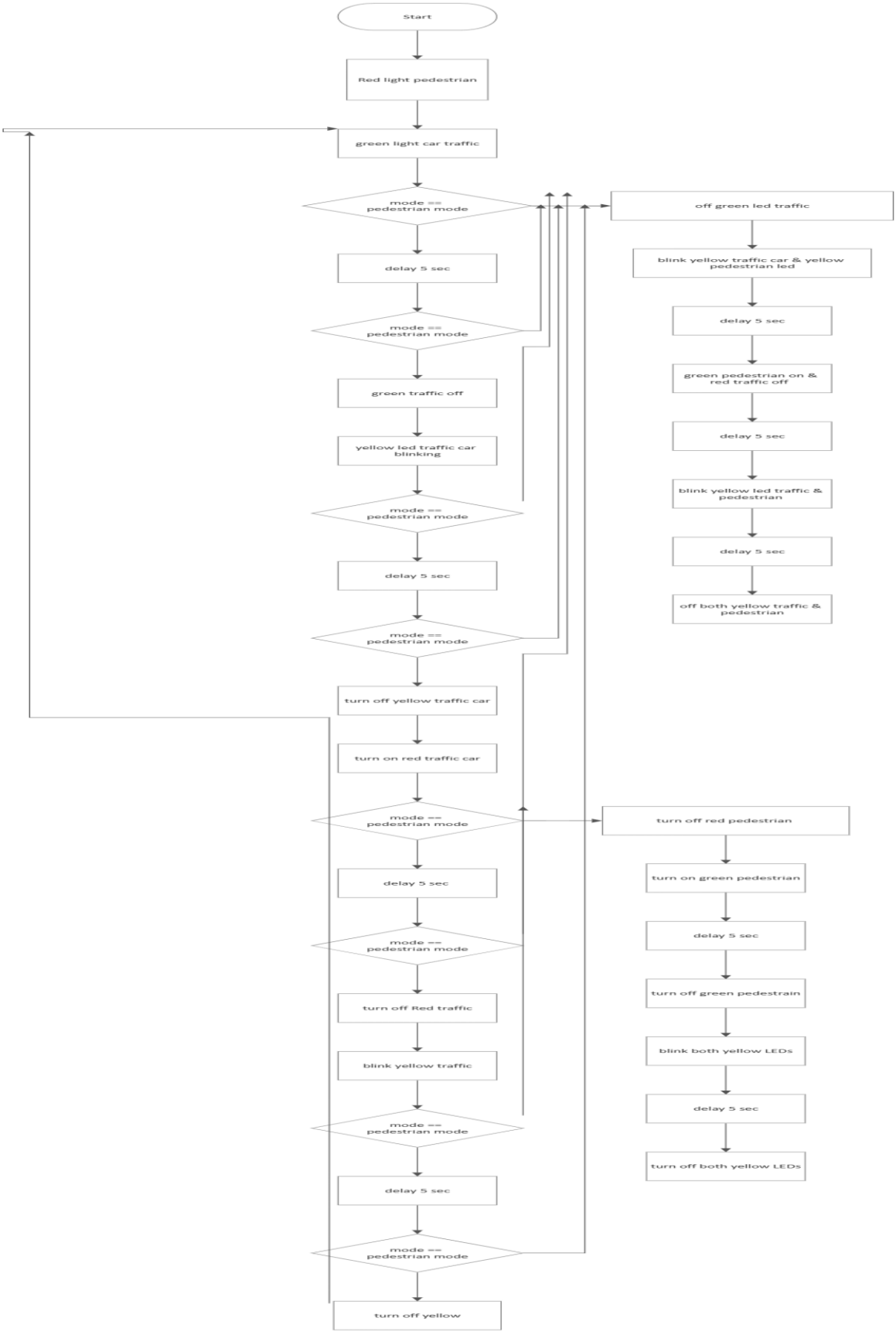
Interrupt APIS

- void EXT_Global_Interrupt();
- void EXT_Interrupt_init();
- void EXT_Enable_Int0();

- **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**


Start
Red light pedestrian
green light car traffic
mode == pedestrian mode
delay 5 sec
mode == pedestrian mode
green traffic off
yellow led traffic car blinking
mode == pedestrian mode
delay 5 sec
mode == pedestrian mode
turn off yellow traffic car
turn on red traffic car
mode == pedestrian mode
delay 5 sec
mode == pedestrian mode
turn off Red traffic
blink yellow traffic
mode == pedestrian mode
delay 5 sec
mode == pedestrian mode
turn off yellow
off green led traffic
blink yellow traffic car & yellow pedestrian led
delay 5 sec
green pedestrian on & red traffic off
delay 5 sec
blink yellow led traffic & pedestrian
delay 5 sec
off both yellow traffic & pedestrian
turn off red pedestrian
turn on green pedestrian
delay 5 sec
turn off green pedestrain
blink both yellow LEDs
delay 5 sec
turn off both yellow LEDs