# System description:

The traffic light system is our normal mode here, it starts with green, blinking yellow then the red which means stop for the cars. We initialize these LEDs through LED driver and DIO driver which gives the pins of the MCU the direction and if each pin takes high (5 volt) or low (0 volt).

The pedestrian has a button that if he/she pressed it, it will trigger an interrupt in the MCU and the LEDs of the pedestrian will ON depends on the status of the car LEDs. The interrupt is function that has a high value than the main function, so, if the ISR (interrupt service routine) comes, the MCU will leave the main function temporarily and will go to serve the ISR then comes back to main function when the interrupt ends.

We have an ISR which is called pedestrian that represents the pedestrian LEDs, when the button is pressed, it triggers the interrupt and starts to see whether the car LED is red, yellow or green.

If its green, then the pedestrian red is ON, and we will blink both car and pedestrian yellow then turn the green LED ON. If its red, then the green LED is ON. If its yellow, then the yellow pedestrian starts blinking and turns the green ON.

Each of these drivers divided into 3 layers in the MCU:

* MCAL Layer

Microcontroller architecture layer, which contains any driver that represents the internal peripheral of the MCU, such as DIO, interrupt, ADC, UART… etc..

* HAL Layer

Hardware architecture layer, which contains any driver that represents the external hardware, such as LED, LCD, Motor or Seven segment.

* Application

Which contains our application and depends on your logic to work your project or program.

# System Design:

Application

[Logic – App driver]

# Chart, box and whisker chart Description automatically generatedSystem flowchart:

Library

Bit math – STD types

MCAL (Microcontroller layer)

DIO – EXTI – GIE drivers

HAL (Hardware layer)

LED driver

# System constrains:

There are no constrains.