1. Define system layers

    - Application
    - ECUAL
    - MCAL
    - UTILITIES

2. Define system drivers

    - DIO
    - LED
    - BUTTON
    - TIMER
    - INTERRUPTS
    - APPLICATION

3. Place each driver into the appropriate layer in the appropriate order

    - APPLICATION LAYER
        - Application driver
            - App.c
            - App.h
    - ECUAL LAYER
        - BUTTON driver
            - button.c
            - button.h
        - LED driver
            - Led.c
            - Led.h
    - MCAL
        - DIO driver
            - dio.c
            - dio.h
        - INTERRUPTS driver
            - interrupts.c
            - interrupts.h
        - TIMER driver
            - timer.c
            - timer.h

    - Utilities

- Registers.h
- Types.h

4. Define APIs that will be used for each driver, with its documentation, description, input arguments, output arguments, and return

1. **Application driver** → has 2 functions

Function prototype	`void init_app(void);`	`void init_app(void);`
description	This function makes all the required initializations for buttons, leds, timers…	This function is used to start program execution (flow of the program)
input arguments, output arguments, and return	It has neither input nor output arguments and it does not return anything	It has neither input nor output arguments and it does not return anything

2. **ECUAL**
   1. Button Driver → has 2 functions

Function prototype	`void BUTTON_init(uint8_t buttonPort, uint8_t buttonPin);`	`void BUTTON_read(uint8_t buttonPort, uint8_t buttonPin, uint8_t* buttonState);`
description	This function takes the port (A/B/C/D), the pin number(0 →7) and it calls `DIO_init` to set the specified pin of the specified port to always input so that the button can be connected to that pin(as button is an input device)	This function takes the port (A/B/C/D), the pin number(0 →7) and calls `DIO_read` which reads the value of the specified pin of the specified port
input arguments, output arguments, and return	It takes `uint8_t buttonPort, uint8_t buttonPin` as inputs parameters. it does not return anything	`uint8_t buttonPort, uint8_t buttonPin, uint8_t* buttonState` as input parameters. it does not return anything

2. LED Driver→ has 4 functions

`void LED_init(uint8_t ledPort,uint8_t ledPin);`	This function takes the port number (A or B or C or D), the pin number and it calls DIO_init to set the specified pin of the specified port to output so that the led can be connected to that pin	It takes uint8_t led_pin ,uint8_t led_port as input parameters and returns nothing
`void LED_ON(uint8_t ledPort,uint8_t ledPin);`	This function takes the port number (A or B or C or D), the pin number and it calls DIO_write to set the specified pin of the specified port to high so that the led can be turned on	It takes uint8_t led_pin ,uint8_t led_port as input parameters and returns nothing
`void LED_OFF(uint8_t ledPort,uint8_t ledPin);`	This function takes the port number (A or B or C or D), the pin number and it calls DIO_write to clear the specified pin of the specified port to low so that the led can be turned off	It takes uint8_t led_pin ,uint8_t led_port as input parameters and returns nothing
`void LED_toggle(uint8_t ledPort,uint8_t ledPin);`	This function takes the port number (A or B or C or D), the pin number and it calls DIO_toggle to toggle the specified pin of the specified port	It takes uint8_t led_pin ,uint8_t led_port as input parameters and returns nothing

3. MCAL layer

1. DIO driver → has 4 functions

`void DIO_init(uint8_t portNumber, uint8_t pinNumber,uint8_t direction); //initializes directions of pins`	This function takes the port number (A or B or C or D), the pin number and whether you want to set it up as input or output and initializes the DIO	It takes uint8_t pinNumber ,uint8_t portNumber ,DIO_DirectionType direction as input parameters and returns nothing
`void DIO_write(uint8_t portNumber, uint8_t pinNumber,uint8_t value); //write data to dio`	This function takes the port number (A or B or C or D), the pin number and whether you want to set it up as high(1) or low(0) and writes on the specified pin of the specified port	It takes uint8_t pinNumber ,uint8_t portNumber ,DIO_DirectionType direction as input parameters and returns nothing
`void DIO_toggle(uint8_t portNumber, uint8_t pinNumber);  //toggle dio`	This function takes the port number (A or B or C or D), the pin number and toggles the value of the specified pin of the specified port	It takes uint8_t uint8_t pinNumber ,uint8_t portNumber as input parameters and returns nothing
`void DIO_read(uint8_t portNumber, uint8_t pinNumber, uint8_t* value);  //read dio`	This function takes the port number (A or B or C or D), the pin number and reads the value of the specified pin of the specified port	It takes uint8_t pinNumber ,uint8_t portNumber , uint8_t *value as input parameters and returns nothing

2. Timer Driver ->has 5 functions

`void timer0_init(void);`	initializing Timer0: what mode and what is the initial value	It has neither input nor output arguments and it does not return anything
`void timer0_start(void);`	starting the timer: setting pre scaler value	It has neither input nor output arguments and it does not return anything

`void timer0_stop(unsigned int NumbOfOverflows);`	checking overflow flag
	
	`while((TIFR &(1<<0))== 0);//busy wait until overflow takes place (bit/flag)==1`
	
	`TCCR0= 0x00;         //timer stop`
		It takes `unsigned int NumbOfOverflows` as input parameter and returns nothing

`void blink_1led(uint8_t Seconds,uint8_t portNumber,uint8_t pinNumber);  //blink function`	Used in blinking the cars' yellow led for whatever seconds I need to	It takes `uint8_t Seconds,uint8_t portNumber,uint8_t pinNumber` as input parameters and returns nothing
`void blink_2leds(uint8_t Seconds,uint8_t portNumber1, uint8_t portNumber2, uint8_t pinNumber1, uint8_t pinNumber2);`	Used in blinking both cars' yellow led and pedestrian's yellow led for whatever seconds I need to	It takes `uint8_t Seconds,uint8_t portNumber1, uint8_t portNumber2, uint8_t pinNumber1, uint8_t pinNumber2` as input parameters and returns nothing

3. Interrupt Driver → has 4 functions

`void enable_global_interrupt(void);`	This function enables global interrupt by setting the 7th bit (I bit) in the sreg register
	
	`SREG|=(1<<7);`
	
	Or simply by calling `sei();` which is also defined in interrupts.h
	
	`#define sei() __asm__ __volatile__ ("sei" ::: "memory")`
		It has no output or input parameters, and it does not return anything

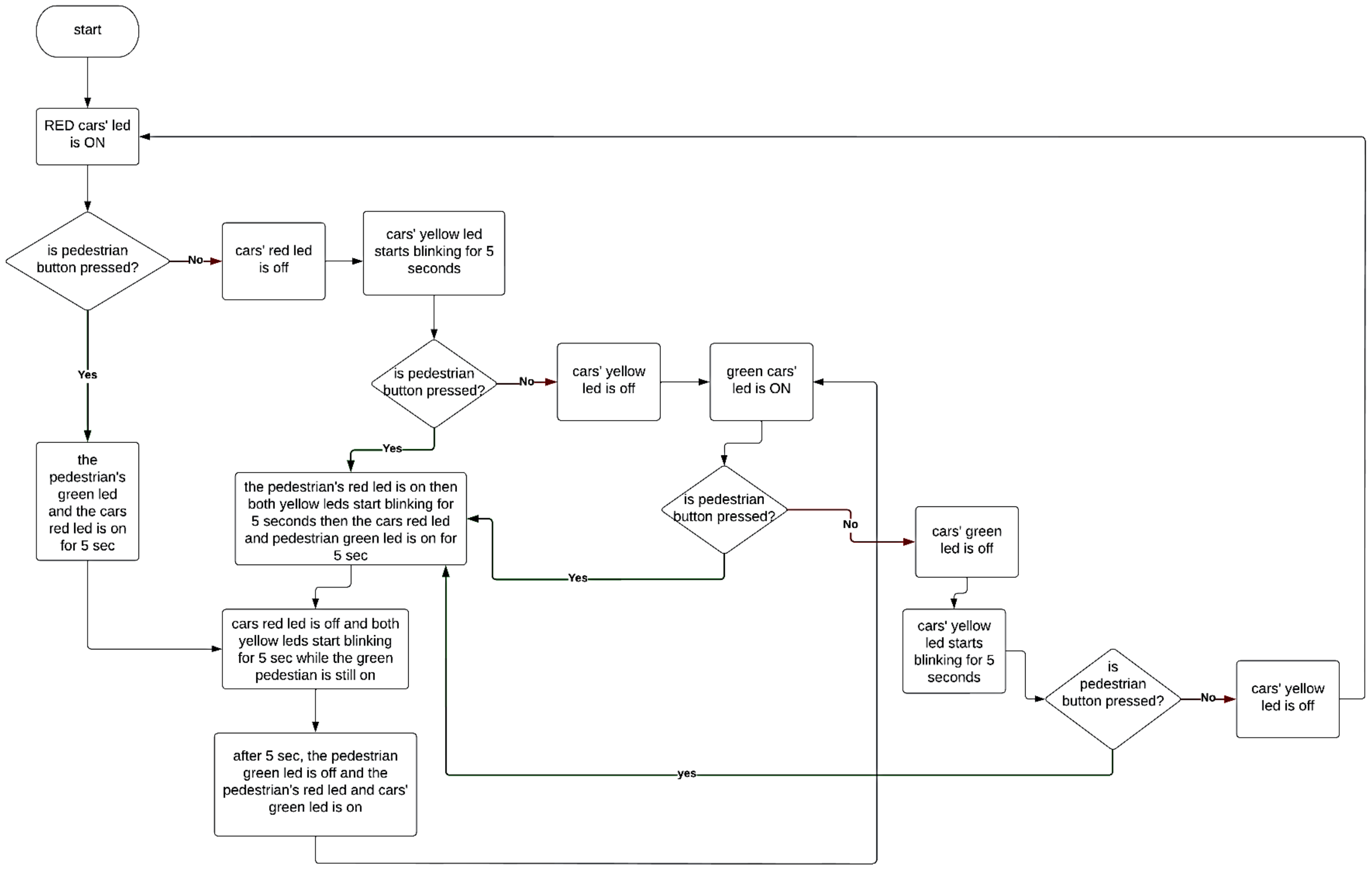
`void disable_global_interrupt(void);`	This function disables global interrupts by calling `cli();` which is also defined in interrupts.h `#define cli() __asm__ __volatile__ ("cli" ::: "memory") // Clear the I-Bit in status register to 0`	It has no output or input parameters, and it does not return anything
`void interrupt_sense_risingEdge (void);`	Choose the external interrupt sense - sense on raising edge. By setting bit0 (ISC00) and bit1(ISC01) which are responsible for interrupt sense control 0 in the MCUCR register `MCUCR |= (1<<1)|(1<<0);`	It has no output or input parameters, and it does not return anything
`void enable_external_INT0(void);`	Enables external interrupt By setting the 6th bit(INT0-external interrupt request 0 enable) in the GICR register `GICR |=(1<<6);`	It has no output or input parameters, and it does not return anything

5. Define the new data types you will use in these drivers

```
typedef unsigned char uint8_t;
```

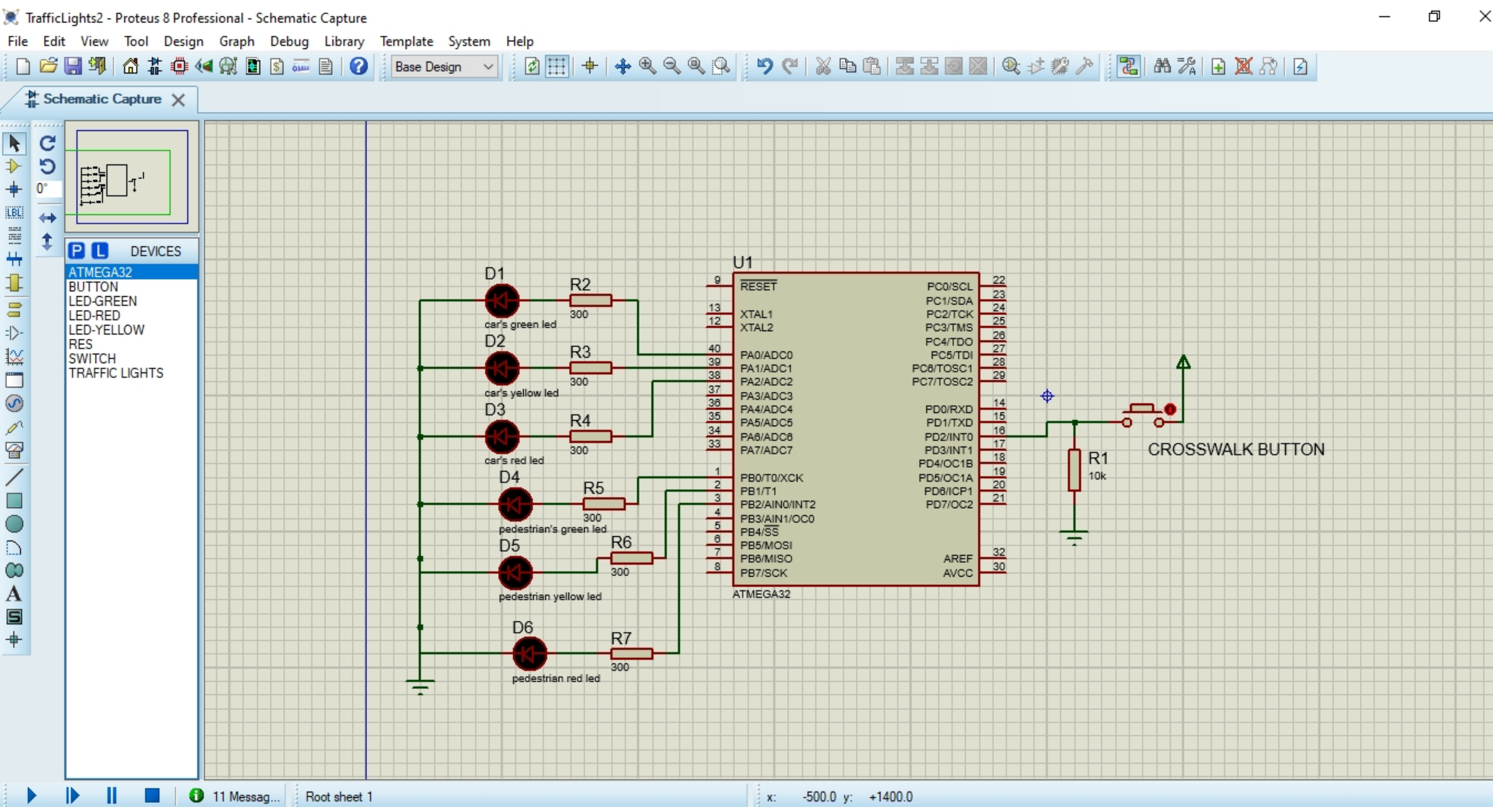
uint8_t is unsigned character whose size is 1 byte ~ 8bits .

system’s flowchart



start
RED cars' led is ON
is pedestrian button pressed?
No
cars' red led is off
cars' yellow led starts blinking for 5 seconds
Yes
the pedestrian's green led and the cars red led is on for 5 sec
is pedestrian button pressed?
No
cars' yellow led is off
green cars' led is ON
Yes
the pedestrian's red led is on then both yellow leds start blinking for 5 seconds then the cars red led and pedestrian green led is on for 5 sec
is pedestrian button pressed?
No
Yes
cars' green led is off
cars red led is off and both yellow leds start blinking for 5 sec while the green pedestian is still on
cars' yellow led starts blinking for 5 seconds
is pedestrian button pressed?
No
cars' yellow led is off
after 5 sec, the pedestrian green led is off and the pedestrian's red led and cars' green led is on
yes

system's design


TrafficLights2 - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Base Design
Schematic Capture
DEVICES
ATMEGA32
BUTTON
LED-GREEN
LED-RED
LED-YELLOW
RES
SWITCH
TRAFFIC LIGHTS
D1
R2
300
car's green led
D2
R3
300
car's yellow led
D3
R4
300
car's red led
D4
R5
300
pedestrian's green led
D5
R6
300
pedestrian yellow led
D6
R7
300
pedestrian red led
U1
ATMEGA32
RESET
XTAL1
XTAL2
PA0/ADC0
PA1/ADC1
PA2/ADC2
PA3/ADC3
PA4/ADC4
PA5/ADC5
PA6/ADC6
PA7/ADC7
PB0/T0/XCK
PB1/T1
PB2/AIN0/INT2
PB3/AIN1/OC0
PB4/SS
PB5/MOSI
PB6/MISO
PB7/SCK
PC0/SCL
PC1/SDA
PC2/TCK
PC3/TMS
PC4/TDO
PC5/TDI
PC6/TOSC1
PC7/TOSC2
PD0/RXD
PD1/TXD
PD2/INT0
PD3/INT1
PD4/OC1B
PD5/OC1A
PD6/ICP1
PD7/OC2
AREF
AVCC
R1
10k
CROSSWALK BUTTON
11 Messag...
Root sheet 1
x: -500.0 y: +1400.0