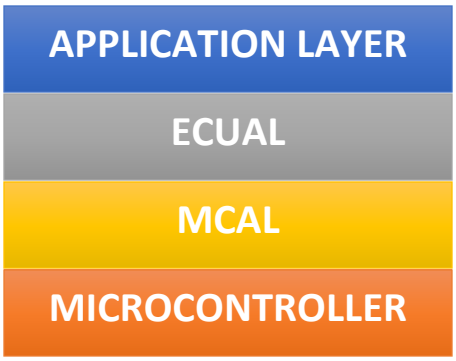
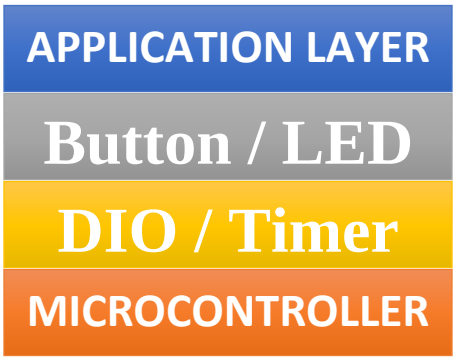
1. Define system layers



2. Define system drivers

   - LED driver
   - Button driver
   - DIO driver
   - Timer driver

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



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

   DIO API

```
void DIO_int(uint8_t portNumber,uint8_t pinNumber , uint8_t direction);
void DIO_write(uint8_t portNumber,uint8_t pinNumber , uint8_t value);
void DIO_toggle(uint8_t portNumber,uint8_t pinNumber );
void DIO_read(uint8_t portNumber,uint8_t pinNumber , uint8_t *value);
```

Timer API

```
void TIMER_init(void);
void TIMER_delay(uint8_t breakDelay);
```

Button API

```
void BUTTOM_init(uint8_t buttomPort,uint8_t buttomPin);
void BUTTOM_read(uint8_t buttomPort,uint8_t buttomPin,uint8_t *value);
```

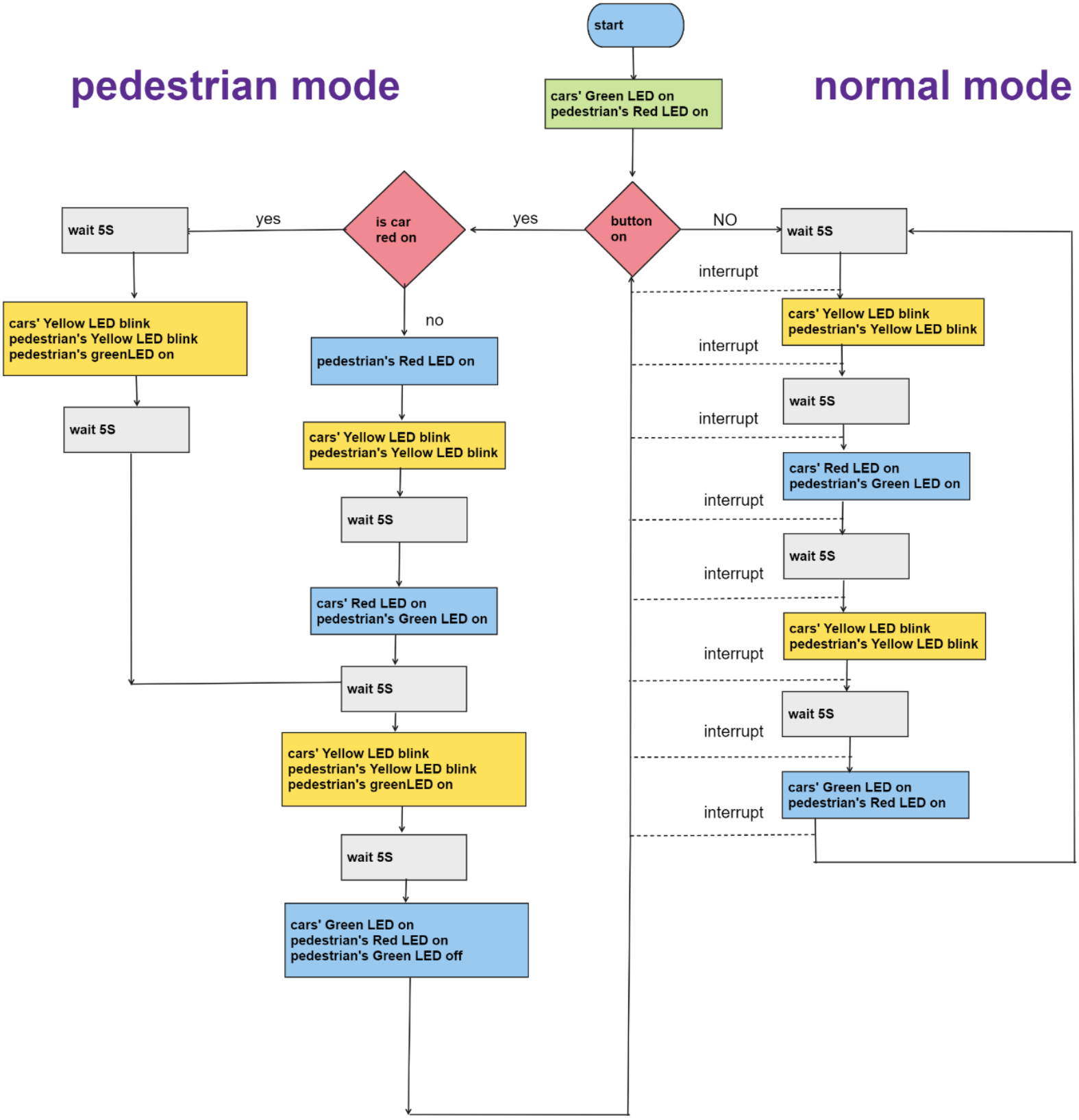
LED API

```
void LED_int(uint8_t ledPort ,uint8_t ledPin);
void LED_on(uint8_t ledPort ,uint8_t ledPin);
void LED_off(uint8_t ledPort ,uint8_t ledPin);
void LED_toggle(uint8_t ledPort ,uint8_t ledPin);
```

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

```
typedef unsigned char uint8_t ;
```

Flow Chart


pedestrian mode
normal mode
start
cars' Green LED on
pedestrian's Red LED on
button on
yes
NO
is car red on
yes
wait 5S
cars' Yellow LED blink
pedestrian's Yellow LED blink
pedestrian's greenLED on
wait 5S
no
pedestrian's Red LED on
cars' Yellow LED blink
pedestrian's Yellow LED blink
wait 5S
cars' Red LED on
pedestrian's Green LED on
wait 5S
cars' Yellow LED blink
pedestrian's Yellow LED blink
pedestrian's greenLED on
wait 5S
cars' Green LED on
pedestrian's Red LED on
pedestrian's Green LED off
wait 5S
interrupt
cars' Yellow LED blink
pedestrian's Yellow LED blink
interrupt
wait 5S
interrupt
cars' Red LED on
pedestrian's Green LED on
interrupt
wait 5S
interrupt
cars' Yellow LED blink
pedestrian's Yellow LED blink
interrupt
wait 5S
interrupt
cars' Green LED on
pedestrian's Red LED on
interrupt