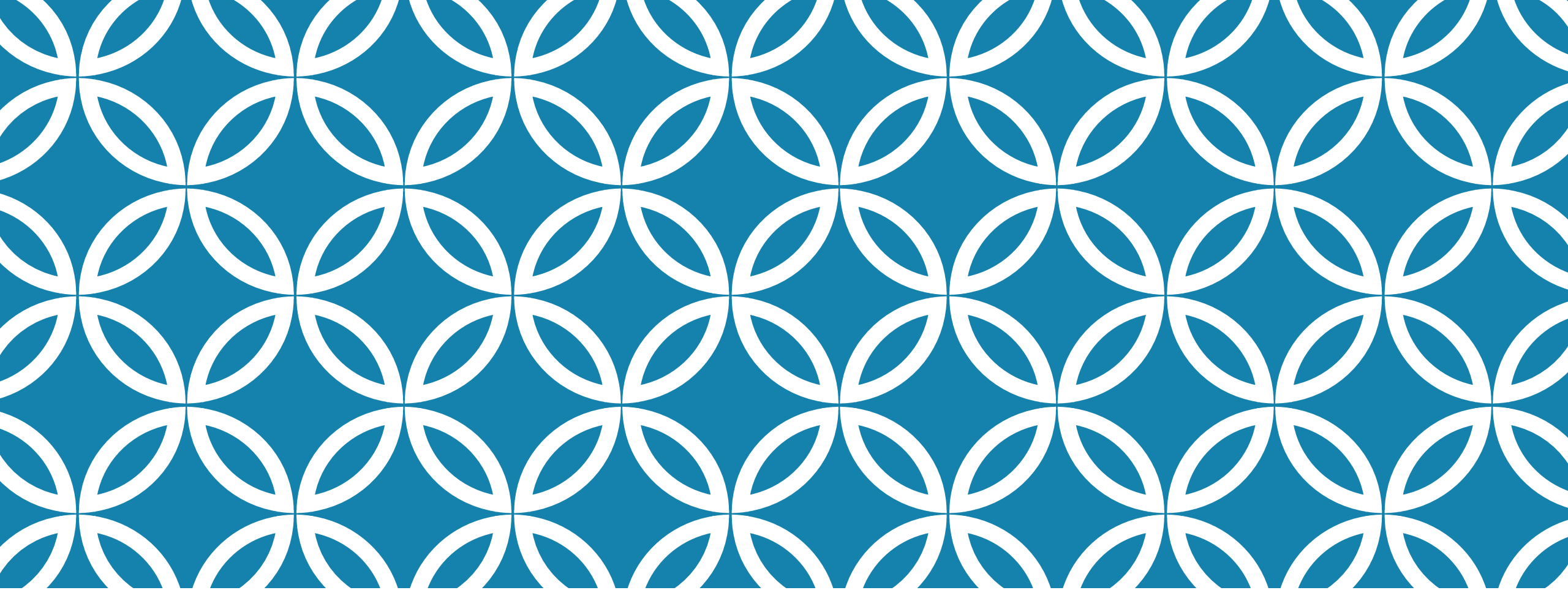
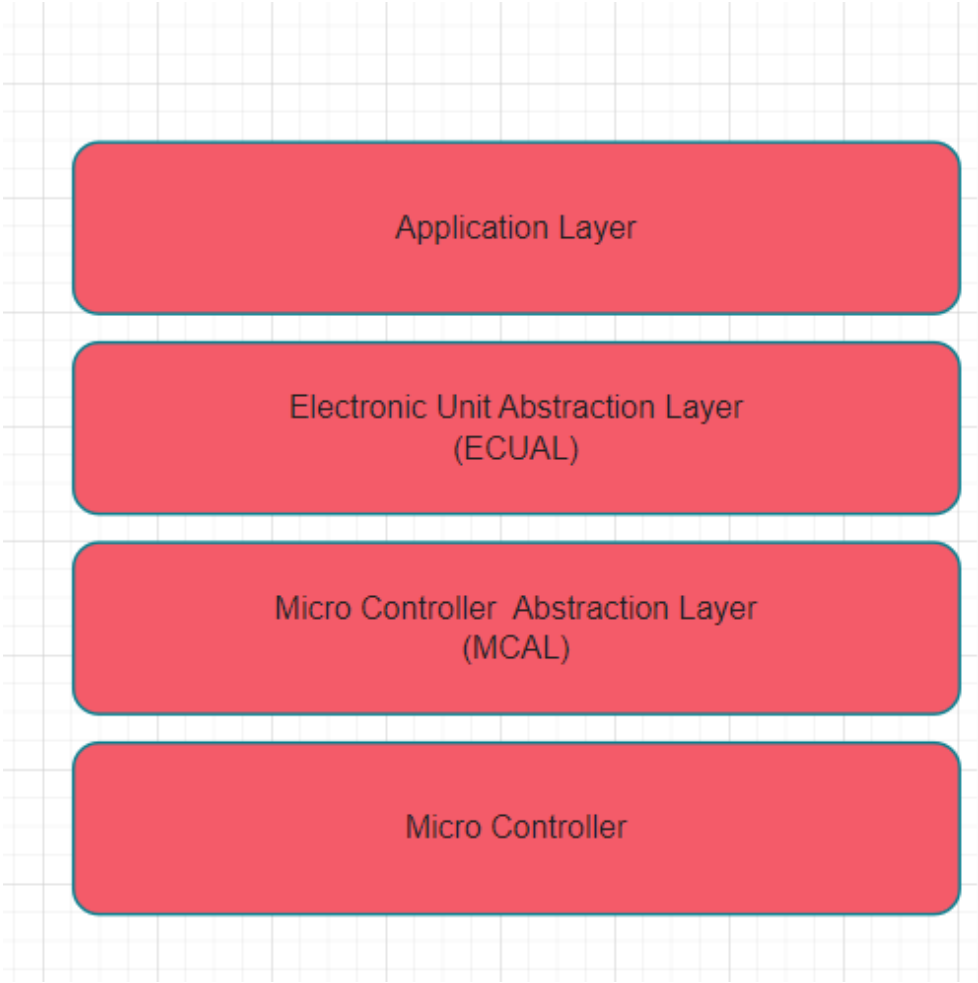


ON DEMAND TRAFFIC LIGHT PROJECT

Embedded Systems

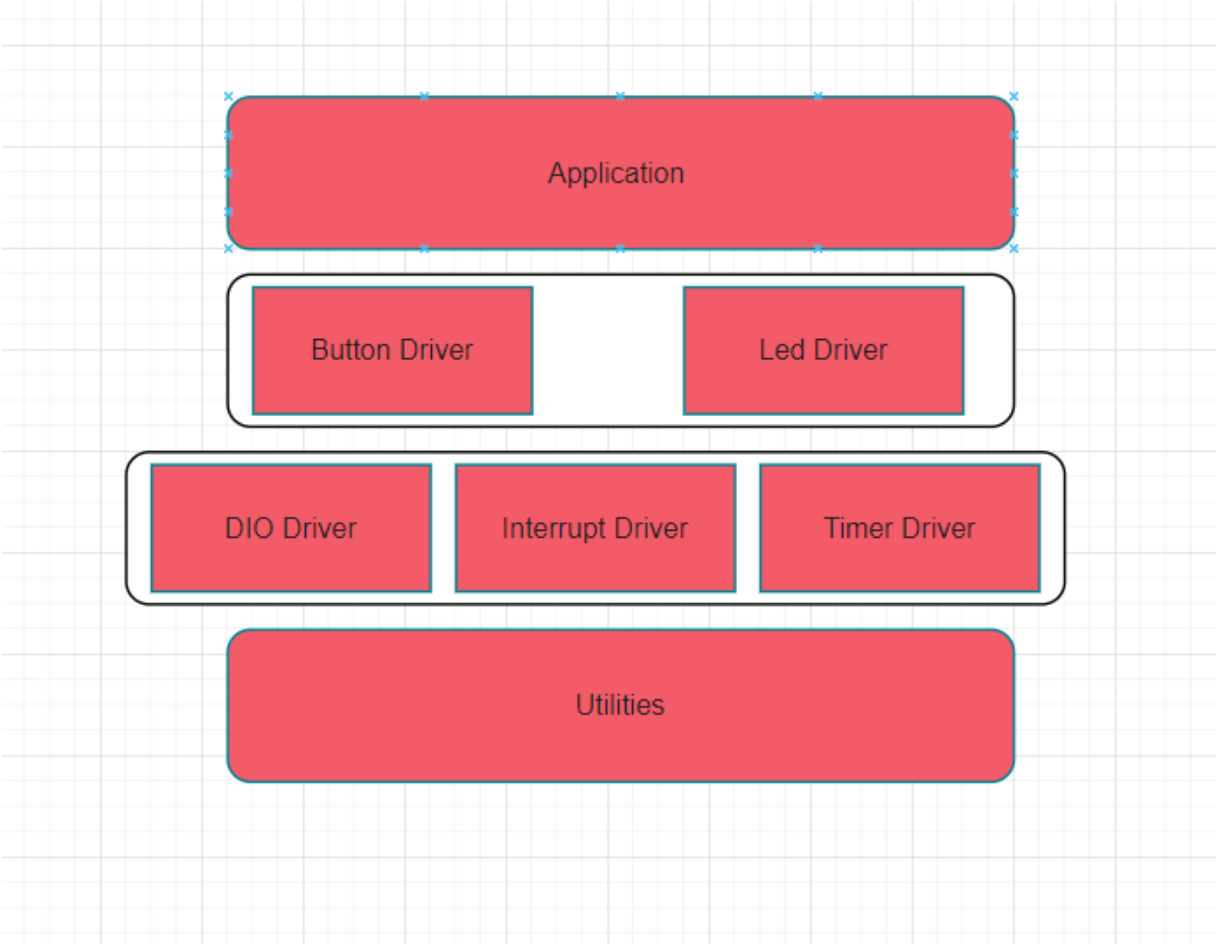
1) SYSTEM DESIGN

"System Layers"



1) SYSTEM DESIGN

"Drivers in Each layer"



2)SYSTEM DESCRIPTION

"Microcontroller Layer"

Contains Mainly Utilities Folder Which Contains (Registers.h) File Defining

1. I/O Registers (Port a ,port b ,port d)
2. Timers Registers.
3. Interrupt Registers.
4. Pin Registers .

2)SYSTEM DESCRIPTION

"MCAL Layer"

Contains Mainly MCAL Folder Which Contains (DIO.C)(DIO.H) Files Inside Dio Driver as:

1. (Dio_init) Is The Initialization of Pins On At mega 32 Device.
2. (Dio_read)Reads Pin Data.
3. (Dio_write)Writes Data On Pin.
4. (Dio_Toggle) Toggles State of Pin Between 0 and 1 .

```
void Dio_init(uint8_t PortNumber ,uint8_t PinNumber ,uint8_t Direction) ;
void Dio_read(uint8_t PortNumber ,uint8_t PinNumber ,uint8_t *Value) ;
void Dio_write(uint8_t PortNumber ,uint8_t PinNumber ,uint8_t Value) ;
void Dio_Toggle(uint8_t PortNumber ,uint8_t PinNumber) ;
```

2)SYSTEM DESCRIPTION

"MCAL Layer"

Contains Mainly MCAL Folder Which Contains (Interrupts.h) Files Inside Dio Driver as:

1. Defining External Interrupts Vectors
2. Macros to Enable or clear global registers
3. ISR macros.

```
#ifndef INCFILE1_H_
#define INCFILE1_H_

#include "../../Utilitis/Registers.h"


#define EXT_INT_0 __vector_1



#define sei() __asm__ __volatile__ ("sei" ::: "memory")

#define Choose_External_Interrupt() MCUCR |= (1 << 0) | (1 << 1);

#define Enable_External_Interrupt() GICR |= (1 << 6);

#define ISR(INT_VECT) \
void INT_VECT(void) __attribute__ ((signal,used)); \
void INT_VECT(void)


#endif /* INCFILE1_H_ */
```

2)SYSTEM DESCRIPTION

"MCAL Layer"

Contains Mainly MCAL Folder Which Contains (Timer.c)(Timer.h) Files Inside Timer Driver as:

1. (Timer_init) Is The Initialization of Timer In Normal.
2. (Timer_Stop)Stops Timer .
3. (Timer0)Starting Timer According to specific prescalar and Number of Overflows .

```
#include "../../Utilitis/Registers.h"
#ifndef TIMER_H_
#define TIMER_H_


void Timer_init (void) ;
void Timer_Stop(void) ;
void Timer0(void) ;


#endif /* TIMER_H_ */
```

2)SYSTEM DESCRIPTION

"ECUAL Layer"

Contains Mainly MCAL Folder Which Contains (Button.c)(Button.h) Files Inside Dio Driver as:

1. (Button_init) Is The Initialization of Button Pins and Port as Input.
2. (Button_read) Checking State Of Button (High – Low) .

```
#ifndef BUTTON_H_
#define BUTTON_H_
#include "../../MCAL/DIO_Driver/Dio.h"

void Button_init(uint8_t ButtonPort,uint8_t ButtonPin);

void Button_read(uint8_t ButtonPort,uint8_t ButtonPin,uint8_t *Value);

#endif /* BUTTON_H_ */
```

2)SYSTEM DESCRIPTION

"ECUAL Layer"

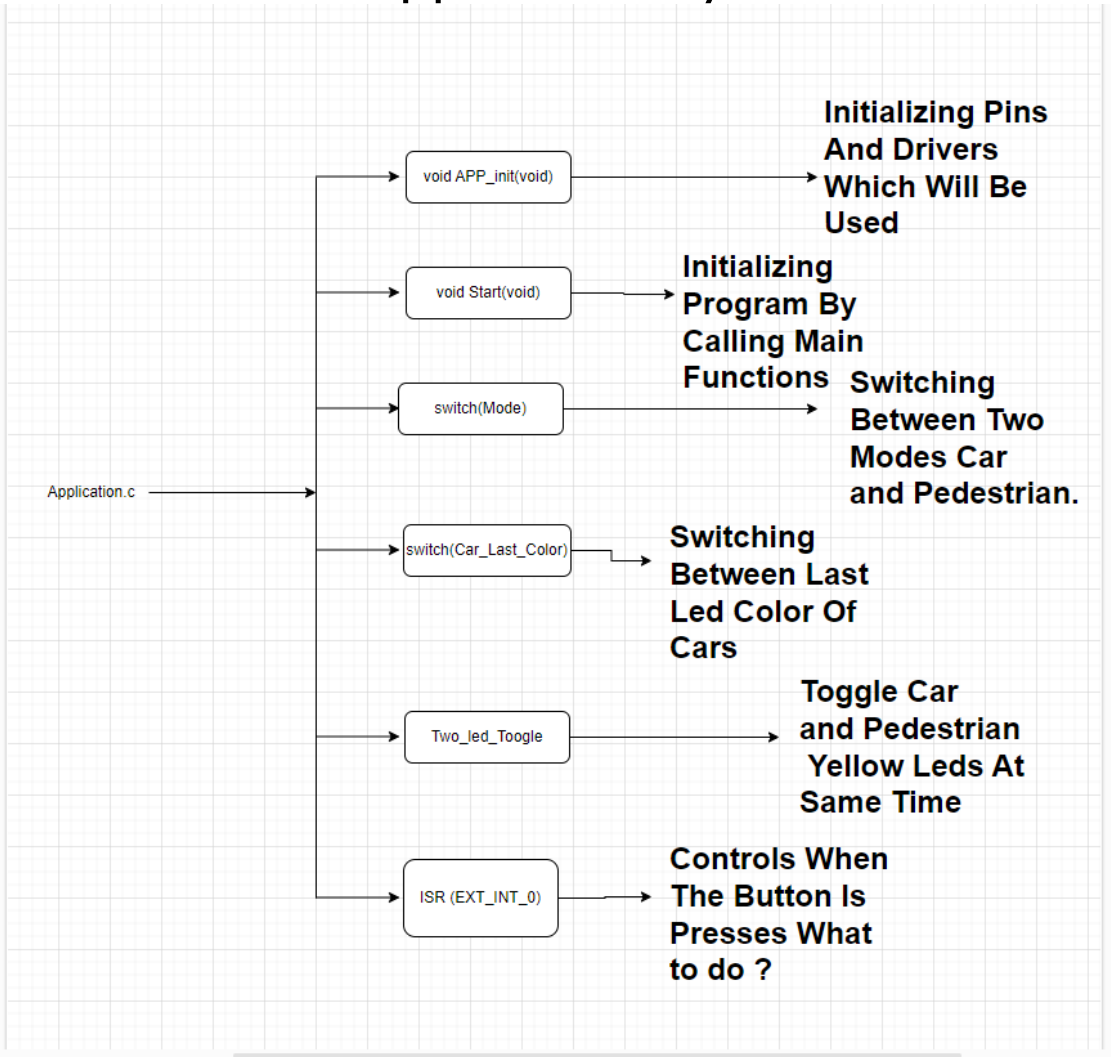
Contains Mainly MCAL Folder Which Contains (Led.c)(Led.h) Files Inside Dio Driver as:

1. (LED_init) Is The Initialization of Led Pins and Port as Output.
2. (LED_On) Writes (High) on pin.
3. (LED_Off) Writes (Low) on pin .
4. (LED_Toggle) Toggle Led pin Between 0 and 1.
5. (Yellow_Led_Blink) Toggle Led pin Between 0 and 1with different Timer Settings

```
void LED_init (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);
void Yellow_Led_Blink (uint8_t LedPort , uint8_t LedPin ) ;
```

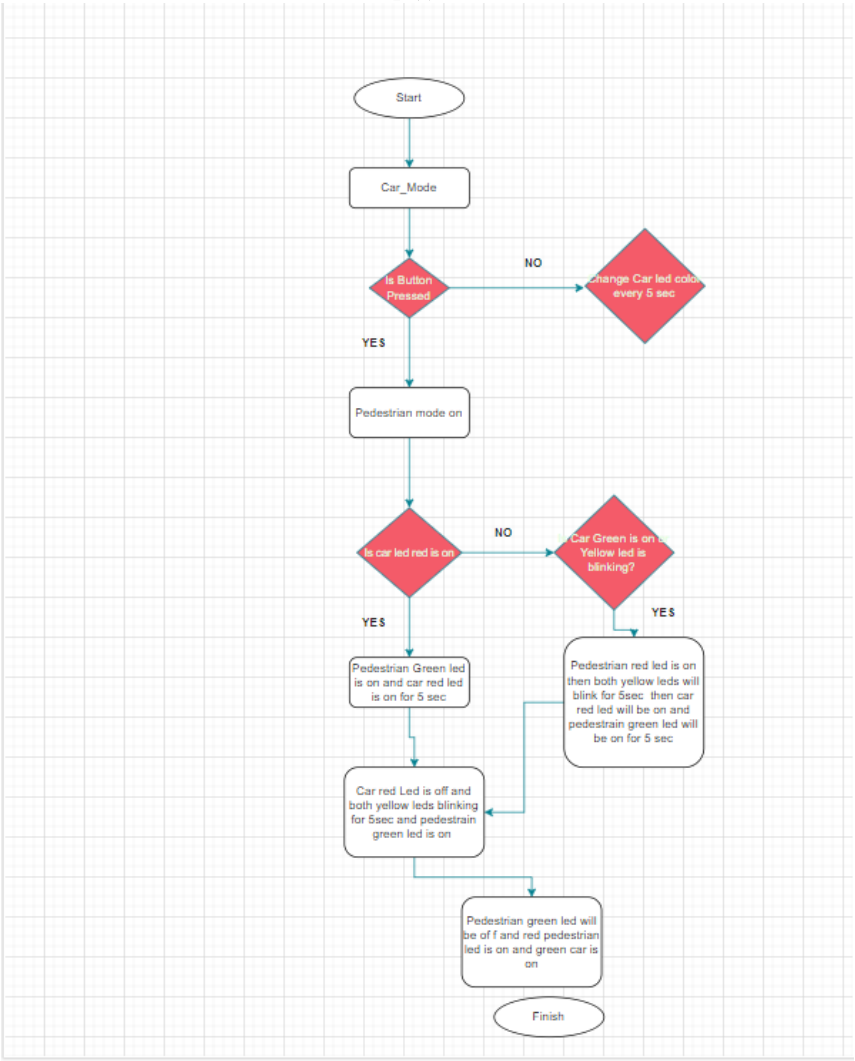
2)SYSTEM DESCRIPTION

"Application Layer"



3)SYSTEM FLOW CHART

“”



4)DESCRIPTION

(App Will Start With Car Mode)

1. Turning off all LEDs.
2. Turn on Car Green LED then Yellow Starts To Blinks then Car red Turn on And So on Until Button Is Pressed .
3. If Button Is Pressed System if Button is on (high or low mode) .
4. System Starts To call Pedestrian Mode To Initialize .
5. System blinks car 's and pedestrian Yellow LED and turn on the pedestrians' Red LED .
6. Turn on Pedestrian Green Led and Car Red Led.
7. Then Both Yellow Leds Will Be Blinking For 5 Sec
8. System Turns on car Green Led and Pedestrian Red Led .
9. System Continue the Pedestrian mode until led cars pass
10. After finishing the pedestrian mode the Car Mode Will Be in Turn .