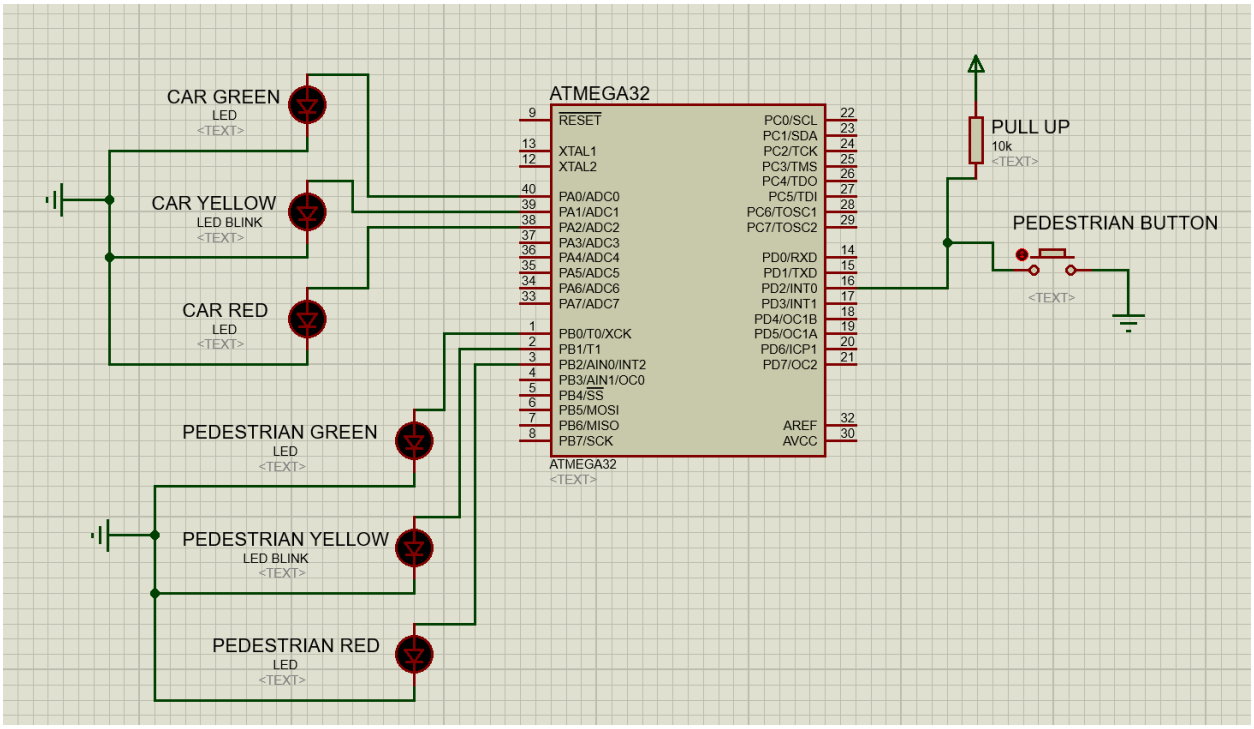
1. Introduction

This document is the Software Requirements Specifications (SRS) and Component Design Documentation (CDD) for the project **On-demand Traffic light control**

2. System Description

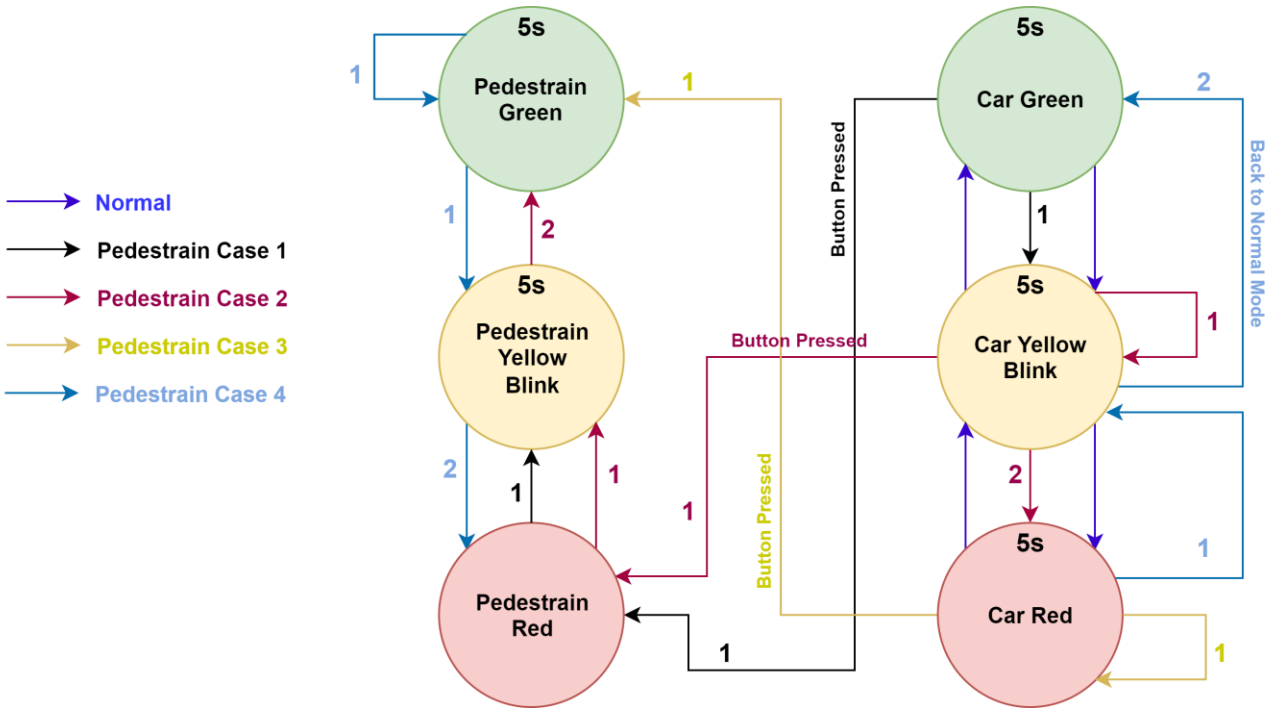
The purpose of this system is to simulate **On-demand Traffic light control.** The system runs on ATmega32 microcontroller and interacts with:

- Pedestrian press button: A button that when pressed simulates changing from Normal Traffic mode to Pedestrian Traffic mode.
- Car Green LED: A LED that acts like a Car Green Traffic Light
- Car Yellow LED: A LED that acts like a Car Yellow Traffic Light
- Car Red LED: A LED that acts like a Car Green Red Light
- Pedestrain Green LED: A LED that acts like a Pedestrain Green Traffic Light
- Pedestrain Yellow LED: A LED that acts like a Pedestrain Yellow Traffic Light
- Pedestrain Red LED: A LED that acts like a Pedestrain Red Traffic Light



3. System's State Machine

The state machine of the system is described in the figure below:



Normal Mode: In this state, the Pedestrain button is not pressed.

Pedestrain Mode: in this state, there are 4 cases:

- Case1: If Pedestrain button is pressed when Car Green is ON, then Pedestrain Red is ON, then Both Car and Pedestrain Yellow is blinking for 5 seconds. Then we go to Case2.
- Case2: If Pedestrain button is pressed when Both Car and Pedestrain Yellow is blinking, then Pedestrain Red is ON, then Both Car Yellow is blinking for 5 seconds. Then we go to Case3.
- Case3: If Pedestrain button is pressed when Car Red is ON, then Pedestrain Green is ON and Car Red is still ON for 5 seconds. Then we go to Case4.
- Case 4: In this case, we return to Normal Mode by set both Car and Pedestrain Yellow to be blinking for 5 seconds, then turn off Car Red and go to Car Green Normal case to start from it again, and put Pedestrain Red to be ON.

4. System Constrains

N/A – there are no system constrains

5. System Design

5.1 System Layers

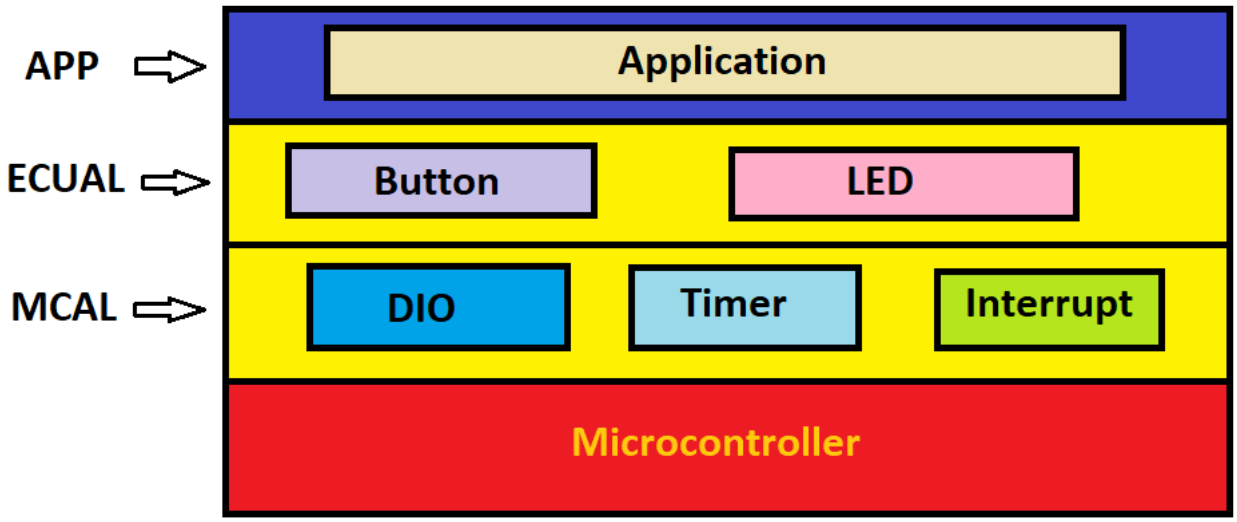
The system consists of **4 layers**:

- Microcontroller layer
- Microcontroller abstraction layer
- ECU abstraction layer
- Application layer

The system has the following **drivers**:

- DIO
- Timer
- Interrupt
- Button
- LED

The system drivers are **distributed** in the layers as follows in the figure below:



5.2 APIs Description

5.2.1 DIO APIs

API Name	errorState_Dio
API Type	Enumeration
Description	Driver State Type
Range	Dio_OK, Dio_Wrong_Port, Dio_Wrong_Pin, Dio_Wrong_Direction, Dio_Wrong_Value

API Name	DIO_init
API Type	Service
Description	Initializes DIO driver
Parameters (in)	uint8t Port, uint8t Pin, uint8t Direction
Parameters (out)	Dio_OK, Dio_Wrong_Port, Dio_Wrong_Pin, Dio_Wrong_Direction, Dio_Wrong_Value
Return	Enumeration errorState_Dio

API Name	DIO_write
API Type	Service
Description	Write on DIO pin
Parameters (in)	uint8t Port, uint8t Pin, uint8t Value
Parameters (out)	Dio_OK, Dio_Wrong_Port, Dio_Wrong_Pin, Dio_Wrong_Direction, Dio_Wrong_Value
Return	Enumeration errorState_Dio

API Name	DIO_read
API Type	Service
Description	Read from DIO pin
Parameters (in)	uint8t Port, uint8t Pin, uint8t * Value
Parameters (out)	Dio_OK, Dio_Wrong_Port, Dio_Wrong_Pin, Dio_Wrong_Direction, Dio_Wrong_Value
Return	Enumeration errorState_Dio

API Name	DIO_toggle
API Type	Service
Description	Toggle DIO pin
Parameters (in)	uint8t Port, uint8t Pin
Parameters (out)	Dio_OK, Dio_Wrong_Port, Dio_Wrong_Pin, Dio_Wrong_Direction, Dio_Wrong_Value
Return	Enumeration errorState_Dio

5.2.2 Timer APIs

API Name	errorState_Timer
API Type	Enumeration
Description	Driver State Type
Range	Timer_OK, Timer_WRONG

API Name	_delay_250m
API Type	Service
Description	Initialize Timer 0 and delay 250ms
Parameters (in)	None
Parameters (out)	Timer_OK, Timer_WRONG
Return	Enumeration errorState_Timer

5.2.3 Interrupt APIs

API Name	errorState_Interrupt
API Type	Enumeration
Description	Driver State Type
Range	Interrupt_OK, Interrupt_WRONG

API Name	Interrupt_init
API Type	Service
Description	Initialize Interrupt 0
Parameters (in)	None
Parameters (out)	Interrupt_OK, Interrupt_WRONG
Return	Enumeration errorState_Interrupt

5.2.4 Button APIs

API Name	errorState_Button
API Type	Enumeration
Description	Driver State Type
Range	Button_OK, Button_Wrong_Port, Button_Wrong_Pin

API Name	BUTTON_init
API Type	Service
Description	Initialize Button Driver
Parameters (in)	uint8t Port, uint8t Pin
Parameters (out)	Button_OK, Button_Wrong_Port, Button_Wrong_Pin
Return	Enumeration errorState_Button

API Name	BUTTON_read
API Type	Service
Description	Read from Button Pin
Parameters (in)	uint8t Port, uint8t Pin, uint8t * Value
Parameters (out)	Button_OK, Button_Wrong_Port, Button_Wrong_Pin
Return	Enumeration errorState_Button

API Name	BUTTON_interrupt
API Type	Service
Description	Read from Button Pin through Interrupt Driver
Parameters (in)	uint8t Port, uint8t Pin
Parameters (out)	Button_OK, Button_Wrong_Port, Button_Wrong_Pin
Return	Enumeration errorState_Button

5.2.5 LED APIs

API Name	errorState_LED
API Type	Enumeration
Description	Driver State Type
Range	LED_OK, LED_Wrong_Port, LED_Wrong_Pin, LED_Wrong_Value

API Name	LED_init
API Type	Service
Description	Initialize LED Driver
Parameters (in)	uint8t Port, uint8t Pin
Parameters (out)	LED_OK, LED_Wrong_Port, LED_Wrong_Pin, LED_Wrong_Value
Return	Enumeration errorState_LED

API Name	LED_on
API Type	Service
Description	Turn ON LED Pin
Parameters (in)	uint8t Port, uint8t Pin
Parameters (out)	LED_OK, LED_Wrong_Port, LED_Wrong_Pin, LED_Wrong_Value
Return	Enumeration errorState_LED

API Name	LED_off
API Type	Service
Description	Turn OFF LED Pin
Parameters (in)	uint8t Port, uint8t Pin
Parameters (out)	LED_OK, LED_Wrong_Port, LED_Wrong_Pin, LED_Wrong_Value
Return	Enumeration errorState_LED

API Name	LED_toggle
API Type	Service
Description	Toggle ON/OFF LED Pin
Parameters (in)	uint8t Port, uint8t Pin
Parameters (out)	LED_OK, LED_Wrong_Port, LED_Wrong_Pin, LED_Wrong_Value
Return	Enumeration errorState_LED

API Name	LED_on_time
API Type	Service
Description	Turn ON LED Pin for a specific time
Parameters (in)	uint8t Port, uint8t Pin, uint8t Value
Parameters (out)	LED_OK, LED_Wrong_Port, LED_Wrong_Pin, LED_Wrong_Value
Return	Enumeration errorState_LED

API Name	LED_blink_time
API Type	Service
Description	Turn ON with blinking LED Pin for a specific time
Parameters (in)	uint8t Port, uint8t Pin, uint8t Value
Parameters (out)	LED_OK, LED_Wrong_Port, LED_Wrong_Pin, LED_Wrong_Value
Return	Enumeration errorState_LED

API Name	LED_double_blink_time
API Type	Service
Description	Turn ON with blinking two different LED Pins for a specific time at the same time
Parameters (in)	uint8t Port1, uint8t Pin1, uint8t Port2, uint8t Pin2, uint8t Value
Parameters (out)	LED_OK, LED_Wrong_Port, LED_Wrong_Pin, LED_Wrong_Value
Return	Enumeration errorState_LED

5.2.6 APP APIs

API Name	System_Mode
API Type	Enumeration
Description	Application State Type
Range	Normal, Pedestrian

API Name	LED_State
API Type	Enumeration
Description	Application State Type
Range	Green, Yellow, Red

API Name	APP_start
API Type	Service
Description	Start Application program
Parameters (in)	None
Parameters (out)	None
Return	None

1.1 New data types Description

1.1.1 DIO new data types

API Name	API Type	Description
input	Symbol	Symbol definition for Input state
output	Symbol	Symbol definition for Output state
Low	Symbol	Symbol definition for Low state
High	Symbol	Symbol definition for High state
A	Symbol	Symbol definition for Port A address
B	Symbol	Symbol definition for Port B address
C	Symbol	Symbol definition for Port C address
D	Symbol	Symbol definition for Port D address
uint8t	Typedef	8-bit unsigned integer data type
uint16t	Typedef	16-bit unsigned integer data type
uint32t	typedef	32-bit unsigned integer data type

1.1.2 Timer new data types

N/A

1.1.3 Interrupt new data types

N/A

1.1.4 Button new data types

N/A

1.1.5 LED new data types

N/A

1.1.6 APP new data types

N/A