On-demand-Traffic-light-control

Contents

Project requirements

implement a traffic lights system with an on-demand crosswalk button.

Crosswalk buttons let the signal operations know that someone is planning to cross the street, so the light adjusts, giving the pedestrian enough time to get across.

System Design

Prepare development environment

Implement the application

Testing the application

Hardware requirements

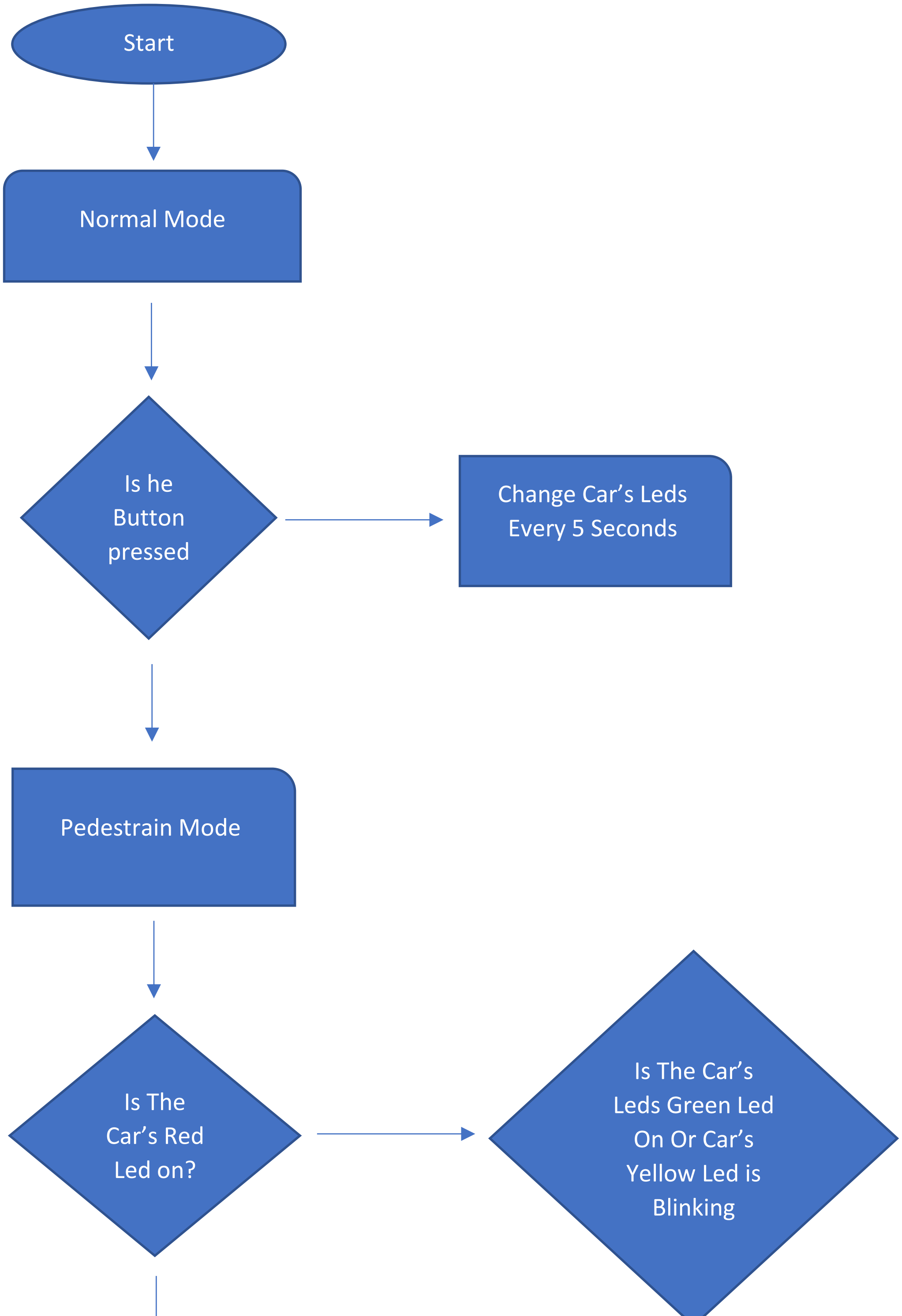
ATmega32 microcontroller

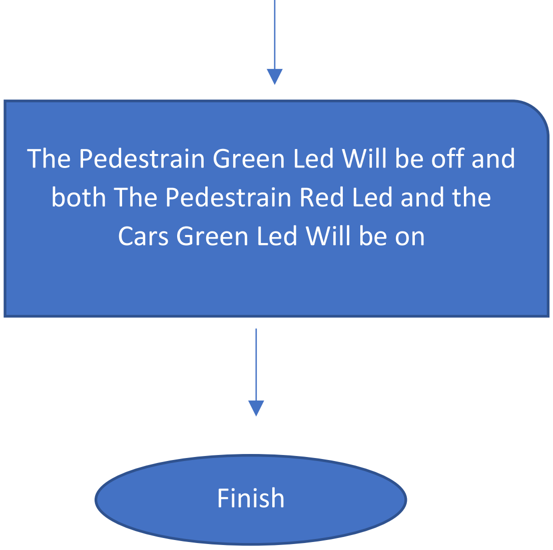
One push button connected to INT0 pin for pedestrian

Three LEDs for cars - Green, Yellow, and Red, connected on port A, pins 0, 1, and 2

Three LEDs for pedestrians - Green, Yellow, and Red, connected on port B, pins 0, 1, and 2

System Design


Start
Normal Mode
Is he Button pressed
Change Car's Leds Every 5 Seconds
Pedestrain Mode
Is The Car's Red Led on?
Is The Car's Leds Green Led On Or Car's Yellow Led is Blinking


The Pedestrain Green Led Will be off and both The Pedestrain Red Led and the Cars Green Led Will be on
Finish

System Design

System Layers

Application
Microcontroler Abstractioin Layer MCAL
Electronic Uint Abstraaction Layer ECUAL
Microcontroler

Drivers in each Layer

<table>
<tr><td colspan="6">Application</td></tr>
<tr><td colspan="3">LED</td><td colspan="3">BUTTON</td></tr>
<tr><td colspan="2">DIO</td><td colspan="2">INTERRUPT</td><td colspan="2">TIMER</td></tr>
<tr><td colspan="6">UTILITIES</td></tr>
</table>

System Description

Microcontroler Layer :

1-Utilities Driver

This driver contain three headers

1-regesters(DIO-Timer-EXT_Interrupt)

2-Types(define important data type)

3-utils(define important mask regesters macros)

MCAL_Layer:

1-DIO_Driver

DIO_init(`void DIO_Init( uint8_t PortNumber , uint8_t PinNumber , uint8_t Direction);`

This function decide the direction of pin in or out

```
void DIO_Read( uint8_t PortNumber , uint8_t PinNumber , uint8_t* value  );
```

This function read data from pin

```
void DIO_Toggle( uint8_t PortNumber , uint8_t PinNumber  );
```

this function toggle the state of the pin

```
void DIO_Write( uint8_t PortNumber , uint8_t PinNumber , uint8_t value  );
```

this function write data on pin

Has important macros to make code readable

2-Timer_Driver:

```
void TIMER_init(void);
```

this function initialize function with normal mode

```
void TIMER_start(uint16_t preScalar, uint32_t number_of_overflow);
```

this function start the timer according to the prescaler and number of overflows

```
void TIMER_stop(void);
```

this function will stop the timer

```
void delay_5s(void);
```

this function will make a delay for five seconds

Has important macros to make code readable

HAL_Layer

1-LED_Driver

```
void LED_Init ( uint8_t LedPort , uint8_t LedPin ) ;
```

this function will call DIO_Init to init special pen

```
void LED_ON ( uint8_t LedPort , uint8_t LedPin )  ;
```

this function will call DIO_Write (High)

```
Init void LED_OFF ( uint8_t LedPort , uint8_t LedPin ) ;
```

this function will call DIO_Write (Low)

```
void LED_TOGGLE ( uint8_t LedPort , uint8_t LedPin ) ;
```

this function will call DIO_Toggle write (High,Low)

Has important macros to make code readable

2-Button_Driver

```
void Button_init(uint8_t buttonPort, uint8_t buttonPin);
```

this function set the direction of the pin as input

```
void Button_get_state(uint8_t buttonPort, uint8_t buttonPin, uint8_t *value);
```

this function check the state of the Button (ON_OFF)

Has important macros to make code readable

Application Layer:

```
void App_init(void);
```

this function will initialize the pins and drivers

```
void App_start(void);
```

this function will start our application by calling function

```
void GreenLED(void);
```

this function will implement the scenario if the cars Green Led is on when the pedestrian push button

```
void RedLED(void);
```

this function will implement the scenario if the cars Red Led is on when the pedestrian push button

```
void YellowLED(void);
```

this function will implement the scenario if the cars Yellow Led is on when the pedestrian push button

```
void PedestrianMode(void);
```

this function will implement pedestrian mode

```
void TurnOffLEDs(void);
```

this function will turn off all leds

```
void toggle_2_LEDs_5s(uint8_t ledPort1, uint8_t ledPin1, uint8_t ledPort2, uint8_t
ledPin2);
```

this function will blink two Leds for 5 seconds

```
void toggle_LED_5s(uint8_t ledPort, uint8_t ledPin);
```

this function will blink one led for 5 seconds