

On damned traffic

Software design

Layer 1

Registers

Typedefs

The screenshot shows the Microchip Studio interface with the registers.h file open in the main editor window. The code defines memory-mapped registers for various ports (PIND, PORTD, PORTC, PORTB) and pins (PIN0, PIN1, etc.). It uses the `register.h` header and provides bit definitions for each pin. The Solution Explorer on the right shows the project structure with files like led.c, led.h, Timer.c, Timer.h, and main.c.

```
#ifndef REGISTERS_H_
#define REGISTERS_H_
//typedefs
typedef unsigned char uint8_t;
typedef unsigned short int16_t;
typedef unsigned int uint32_t;
typedef signed int int32_t;
typedef long unsigned int uint32z_t;
typedef long signed int int32z_t;

//port d
#define PIND *((volatile uint8_t*)0x30)
#define DIND *((volatile uint8_t*)0x31)
#define PORTD *((volatile uint8_t*)0x22)

//port c
#define PINC *((volatile uint8_t*)0x33)
#define DINC *((volatile uint8_t*)0x34)
#define PORTC *((volatile uint8_t*)0x35)

//port b
#define PINB *((volatile uint8_t*)0x00)
#define DIB *((volatile uint8_t*)0x01)
#define PORTB *((volatile uint8_t*)0x02)
83 %
```

Layer 2

dio device

The screenshot shows the Microchip Studio interface with the dio.h file open in the main editor window. The code defines ports A, B, C, and D, along with IN and OUT states and HIGH/LOW levels. It includes functions for initializing, writing, toggling, and reading pins. The Solution Explorer on the right shows the project structure with files like led.c, led.h, Timer.c, Timer.h, and main.c.

```
//defining ports
#define PORT_A 'A'
#define PORT_B 'B'
#define PORT_C 'C'
#define PORT_D 'D'
// defining in and out
#define IN 0
#define OUT 1
//defining high and low
#define High 1
#define Low 0

void DIO_init(unit8_t portNum,unit8_t pinNum,unit8_t direction);
void DIO_write(unit8_t portNum,unit8_t pinNum,unit8_t value);
void DIO_toggle(unit8_t portNum,unit8_t pinNum);
void DIO_read(unit8_t portNum,unit8_t pinNum,unit8_t *value);

#endif /* DIO_H_ */
```

The screenshot shows the Microchip Studio interface. The main window displays the C code for a DIO driver. The Solution Explorer on the right lists project files including dio.c, dio.h, LED Driver, and Timer Driver.

```
main.c registers.h dio.c led.h led.c Timer.c Timer.h

dio.c
}
break;

case PORT_D:
    if (direction==D)
        DORD &= ~(1<<pinNum);
    else if (direction==OUT)
    {
        DORD |=(1<<pinNum);
    }
    else
    {
        //error handing
    }
    break;
}

void DIO_write(unit8_t portNum,unit8_t pinNum,unit8_t value){
    switch(portNum)
    {
        case PORT_A:
            if (value==low){
                PORTA &= ~(1<<pinNum);
            }
            else{
                PORTA |= (1<<pinNum);
            }
    }
}
```

The screenshot shows the Microchip Studio IDE interface. The title bar reads "My first project - Microchip Studio". The menu bar includes File, Edit, View, VASSISTX, ASF, Project, Build, Debug, Tools, Window, Help. The toolbar has icons for New, Open, Save, Build, Run, Stop, and others. The main window shows a code editor with C code for a DIO driver. The code includes includes for stdio.h, dio.h, led.h, and Timer.h, and defines a DIO_init function that sets up port pins as inputs or outputs. The Solution Explorer on the right lists the project structure with files like dio.c, led.c, and timer.c. The Properties window is also visible.

```
main.h
registers.h
dio.h
dio.c
led.h
led.c
Timer.h
Timer.c

//include dio.h
#include "dio.h"
#include "led.h"
#include "Timer.h"

void DIO_init(uint8_t portNum, uint8_t pinNum, uint8_t direction)
{
    switch(portNum)
    {
        case PORT_A:
            if (direction==IN)
            {
                DDRA &=~ (1<<pinNum);
            }
            else if (direction==OUT)
            {
                DDRA |= (1<<pinNum);
            }
        else
        {
            //error handing
        }
        break;
    case PORT_B:
        if (direction==IN)
        {
            DDRB &=~ (1<<pinNum);
        }
        else if (direction==OUT)
        {
            DDRB |= (1<<pinNum);
        }
    }
}
```

Layer 3 timer

The screenshot shows the Microchip Studio IDE interface. The main window displays the code for `TIMER0.h`. The code defines a prescaler enum with values 1, 8, 64, 256, and 1024. It also includes prototypes for `init`, `start`, `setPreload`, `stop`, `reset`, and `delay` functions. The Solution Explorer on the right shows the project structure with files like `LED.h`, `TRAFFIC_LIGHTS.c`, `EXT_INTR.c`, `main.c`, and `TIMER0.c`. The Error List at the bottom shows 0 errors, 0 warnings, and 0 messages.

```
#ifndef TIMER0_H
#define TIMER0_H
// enum for prescaler choice
typedef enum
{
    NoClkSource = 0,
    Prescaler_1,
    Prescaler_8,
    Prescaler_64,
    Prescaler_256,
    Prescaler_1024
} Prescaler;
// functions prototypes
void TIMER0_init(Prescaler prescaler, uint8_t preload);
void TIMER0_start(int N, uint8_t preload);
void TIMER0_setPreload(uint8_t preload);
void TIMER0_stop(void);
void TIMER0_reset(void);
void TIMER0_delay(uint16_t TimeMs);

#endif /* TIMER0_H */
```

The screenshot shows the Microchip Studio IDE interface. The top menu bar includes File, Edit, View, VASSIOT, ASF, Project, Build, Debug, Tools, Window, Help, Advanced Mode, and Quick Launch (Ctrl+Q). The toolbar contains icons for Open, Save, Build, Run, Stop, and Debug. The main editor window displays C code for the TIMER0 module, specifically for setting up traffic lights. The code includes functions for setting preload, starting, stopping, and incrementing a counter. The Solution Explorer on the right shows the project structure with files like LED.h, TRAFFIC_LIGHTS.c, and main.c. The Properties and Error List tabs are also visible at the bottom.

```
#include "TIMERO.h"
#include "TIMEROc.h"
#include "TRAFFIC_LIGHTS.h"
#include "EXT_INT.h"
#include "EXT_INT.c"
#include "app.h"
#include "app.c"
#include "LED.h"
#include "Led.c"

// busy wait until timer overflows
while (Get_Bit(TIFR_REG, 0) == 0)
{
    ;
    // increment counter
    N_counter++;
    // set timer preload
    TIMER0_setPreload(preload);
    // clear ovf flag by setting to the flag
    Set_Bit(TIFR_REG, 0);
}

void TIMER0_setPreload(uint8_t preload)
{
    // set preload to TCNT register
    TCNT0_REG = preload;
}

void TIMER0_stop(void)
{
    // remove clock source to stop timer
    TCCR0_REG = 0;
}
```

Layer 4 : external interrupt

On-demand Traffic light control - Microchip Studio

File Edit View VASSISTX ASF Project Build Debug Tools Window Help

Debug Browser Almega32 None on

TIME0.h TIMER0.c TRAFFIC_LIGHTS.h TRAFFIC_LIGHTS.c EXT_INT.h EXT_INT.c app.h app.c LED.h Led.c

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

// enum for interrupt sense control

=typedef enum

{

LowLevel = 0,

Anychange,

Fallingedge,

Risingedge

} SenseControl;

// enum for interrupt choice

=typedef enum

{

INT0 = 0,

INT1,

INT2

} Interrupt;

// functions prototypes

void EXT_INT_init(Interrupt interrupt, SenseControl senseControl);

void EXT_INT_enable(Interrupt interrupt);

110%

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense

Description

VA Find References Results Error List

Output

Ready

Type here to search

Windows Taskbar

Advanced Mode Quick Launch (Ctrl+Q)

Ln 35 Col 1 Ch 1 INS

16°C 10:13 PM 2/22/2023

On-demand Traffic light control - Microchip Studio

File Edit View VASSISTX ASF Project Build Debug Tools Window Help

Debug Browser Almega32 None on

TIME0.h TIMER0.c TRAFFIC_LIGHTS.h TRAFFIC_LIGHTS.c EXT_INT.h EXT_INT.c app.h app.c LED.h Led.c

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

// enum for interrupt sense control

=typedef enum

{

LowLevel = 0,

Anychange,

Fallingedge,

Risingedge

} SenseControl;

// enum for interrupt choice

=typedef enum

{

INT0 = 0,

INT1,

INT2

} Interrupt;

// functions prototypes

void EXT_INT_init(Interrupt interrupt, SenseControl senseControl);

void EXT_INT_enable(Interrupt interrupt);

110%

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense

Description

VA Find References Results Error List

Output

Ready

Type here to search

Windows Taskbar

Advanced Mode Quick Launch (Ctrl+Q)

Ln 35 Col 1 Ch 1 INS

16°C 10:13 PM 2/22/2023

On-demand Traffic light control - Microchip Studio

File Edit View VASSISTX ASF Project Build Debug Tools Window Help

Debug Browser Almega32 None on

TIME0.h TIMER0.c TRAFFIC_LIGHTS.h TRAFFIC_LIGHTS.c EXT_INT.h EXT_INT.c app.h app.c LED.h Led.c

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

// enum for interrupt sense control

=typedef enum

{

LowLevel = 0,

Anychange,

Fallingedge,

Risingedge

} SenseControl;

// enum for interrupt choice

=typedef enum

{

INT0 = 0,

INT1,

INT2

} Interrupt;

// functions prototypes

void EXT_INT_init(Interrupt interrupt, SenseControl senseControl);

void EXT_INT_enable(Interrupt interrupt);

110%

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense

Description

VA Find References Results Error List

Output

Ready

Type here to search

Windows Taskbar

Advanced Mode Quick Launch (Ctrl+Q)

Ln 35 Col 1 Ch 1 INS

16°C 10:13 PM 2/22/2023

Activate Windows

Go to Settings to activate Windows.

Solution Explorer

Search Solution Explorer (Ctrl+U)

- LED.h
- TRAFFIC_LIGHTS
 - TRAFFIC_LIGHTS.c
 - TRAFFIC_LIGHTS.h
- middle layer
 - DIO DEVICE
 - DIO.c
 - DIO.h
 - EXTERNAL_INTERRUPT
 - EXT_INT.h
 - EXT_INT.c
 - TIMER0_NORMAL
 - TIMER0.c
 - TIMER0.h
 - lower layer
 - main.c

VA View VA Outline Solution Explorer

Properties

Search Error List

Layer 5: led

On-demand Traffic light control - Microchip Studio

File Edit View VASSISTX ASF Project Build Debug Tools Window Help

Debug Browser Almega32 None on

LED.h

```
/*
 * LED.h
 *
 * * Created: 22/02/2023
 * * Author: osama
 */
#ifndef LED_H_
#define LED_H_
// Functions prototypes
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);
#endif /* LED_H_ */
```

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense

Description

VA Find References Results Error List

Output

Ready

Type here to search

On-demand Traffic light control - Microchip Studio

File Edit View VASSISTX ASF Project Build Debug Tools Window Help

Debug Browser Almega32 None on

Led.c

```
#include "LED.h"
void LED_init(uint8_t ledPin, uint8_t ledPort)
{
    // initialize led pin to be output
    DIO_init(ledPin, ledPort, OUTPUT);
}
void LED_on(uint8_t ledPin, uint8_t ledPort)
{
    // set pin value high
    DIO_write(ledPin, ledPort, HIGH);
}
void LED_off(uint8_t ledPin, uint8_t ledPort)
{
    // set pin value low
    DIO_write(ledPin, ledPort, LOW);
}
void LED_toggle(uint8_t ledPin, uint8_t ledPort)
{
    // toggle pin
    DIO_toggle(ledPin, ledPort);
}
```

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense

Description

VA Find References Results Error List

Output

Ready

Type here to search

Activate Windows

Go to Settings to activate Windows.

Ln 1 Col 1 Ch 1 INS

Advanced Mode Quick Launch (Ctrl+Q)

Solution Explorer

Search Solution Explorer (Ctrl+U)

- application
 - app.c
 - app.h
- led_trafficLayer
 - LED
 - Led.c
 - LED.h
 - TRAFFIC_LIGHTS
 - TRAFFIC_LIGHTS.c
 - TRAFFIC_LIGHTS.h
- middle_layer
 - DIO_DEVICE
 - DIO.c
 - DIO.h
 - EXTERNAL_INTERRUPT
 - EXT_INT.c

VA View VA Outline Solution Explorer Properties

Search Error List

Project File Line

Activate Windows

Go to Settings to activate Windows.

Ln 1 Col 1 Ch 1 INS

Advanced Mode Quick Launch (Ctrl+Q)

Solution Explorer

Search Solution Explorer (Ctrl+U)

- application
 - app.c
 - app.h
- led_trafficLayer
 - LED
 - Led.c
 - LED.h
 - TRAFFIC_LIGHTS
 - TRAFFIC_LIGHTS.c
 - TRAFFIC_LIGHTS.h
- middle_layer
 - DIO_DEVICE
 - DIO.c
 - DIO.h
 - EXTERNAL_INTERRUPT
 - EXT_INT.c

VA View VA Outline Solution Explorer Properties

Search Error List

Project File Line

Activate Windows

Go to Settings to activate Windows.

Ln 1 Col 1 Ch 1 INS

Advanced Mode Quick Launch (Ctrl+Q)

Solution Explorer

Search Solution Explorer (Ctrl+U)

- application
 - app.c
 - app.h
- led_trafficLayer
 - LED
 - Led.c
 - LED.h
 - TRAFFIC_LIGHTS
 - TRAFFIC_LIGHTS.c
 - TRAFFIC_LIGHTS.h
- middle_layer
 - DIO_DEVICE
 - DIO.c
 - DIO.h
 - EXTERNAL_INTERRUPT
 - EXT_INT.c

VA View VA Outline Solution Explorer Properties

Search Error List

Project File Line

Layer 6: full traffic light control

On-demand Traffic light control - Microchip Studio

File Edit View VASSISTX ASF Project Build Debug Tools Window Help

Debug Browser None on

Solution Explorer

Search Solution Explorer (Ctrl+U)

- application
 - app.c
 - app.h
- led_trafficlayer
 - LED
 - Led.c
 - Led.h
 - TRAFFIC_LIGHTS
 - TRAFFIC_LIGHTS.c
 - TRAFFIC_LIGHTS.h
- middle_layer
 - DIO DEVICE
 - DIO.c
 - DIO.h
 - EXTERNAL_INTERRUPT
 - EXT_INT.c

VA View VA Outline Solution Explorer

Properties

TRAFFIC_LIGHTS.h File Properties

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense

Description

VA Find References Results Error List

Output

Ready

Type here to search

Windows Taskbar

Activate Windows Go to Settings to activate Windows.

Advanced Mode Quick Launch (Ctrl+Q)

Ln 1 Col 1 Ch 1 INS

16°C 10:14 PM 2/22/2023

On-demand Traffic light control - Microchip Studio

File Edit View VASSISTX ASF Project Build Debug Tools Window Help

Debug Browser None on

Solution Explorer

Search Solution Explorer (Ctrl+U)

- application
 - app.c
 - app.h
- led_trafficlayer
 - LED
 - Led.c
 - Led.h
 - TRAFFIC_LIGHTS
 - TRAFFIC_LIGHTS.c
 - TRAFFIC_LIGHTS.h
- middle_layer
 - DIO DEVICE
 - DIO.c
 - DIO.h
 - EXTERNAL_INTERRUPT
 - EXT_INT.c

VA View VA Outline Solution Explorer

Properties

TRAFFIC_LIGHTS.h File Properties

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense

Description

VA Find References Results Error List

Output

Ready

Type here to search

Windows Taskbar

Activate Windows Go to Settings to activate Windows.

Advanced Mode Quick Launch (Ctrl+Q)

Ln 1 Col 1 Ch 1 INS

16°C 10:14 PM 2/22/2023

On-demand Traffic light control - Microchip Studio

Solution Explorer

- application
 - app.c
 - app.h
 - led_trafficLayer
 - LED
 - Led.c
 - Led.h
 - TRAFFIC_LIGHTS
 - TRAFFIC_LIGHTS.c
 - TRAFFIC_LIGHTS.h
 - middle_layer
 - DIO_DEVICE
 - DIO.c
 - DIO.h
 - EXTERNAL_INTERRUPT
 - EXT_INT.c
 - EXT_INT.h

Error List

Entire Solution | 0 Errors | 0 Warnings | 0 Messages | Build + IntelliSense

Description

VA Find References Results | Error List

Output

Ready

Type here to search

Windows Taskbar

Activate Windows
Go to Settings to activate Windows.

16°C 10:15 PM 2/22/2023

On-demand Traffic light control - Microchip Studio

Solution Explorer

- application
 - app.c
 - app.h
 - led_trafficLayer
 - LED
 - Led.c
 - Led.h
 - TRAFFIC_LIGHTS
 - TRAFFIC_LIGHTS.c
 - TRAFFIC_LIGHTS.h
 - middle_layer
 - DIO_DEVICE
 - DIO.c
 - DIO.h
 - EXTERNAL_INTERRUPT
 - EXT_INT.c

Error List

Entire Solution | 0 Errors | 0 Warnings | 0 Messages | Build + IntelliSense

Description

VA Find References Results | Error List

Output

Ready

Type here to search

Windows Taskbar

Activate Windows
Go to Settings to activate Windows.

16°C 10:15 PM 2/22/2023

On-demand Traffic light control - Microchip Studio

Solution Explorer

- application
 - app.c
 - app.h
 - led_trafficLayer
 - LED
 - Led.c
 - Led.h
 - TRAFFIC_LIGHTS
 - TRAFFIC_LIGHTS.c
 - TRAFFIC_LIGHTS.h
 - middle_layer
 - DIO_DEVICE
 - DIO.c
 - DIO.h
 - EXTERNAL_INTERRUPT
 - EXT_INT.c

Error List

Entire Solution | 0 Errors | 0 Warnings | 0 Messages | Build + IntelliSense

Description

VA Find References Results | Error List

Output

Ready

Type here to search

Windows Taskbar

Activate Windows
Go to Settings to activate Windows.

16°C 10:15 PM 2/22/2023

On-demand Traffic light control - Microchip Studio

File Edit View VASSISTX ASF Project Build Debug Tools Window Help

Advanced Mode Quick Launch (Ctrl+Q)

TIMER0.h TIMER0.c TRAFFIC_LIGHTS.h TRAFFIC_LIGHTS.c EXT_INT.h EXT_INT.c app.h app.c LED.h Led.c

CarsGreenEnable

```
void CarsGreenEnable(void)
{
    // toggle cars yellow
    LED_toggle(PIN1, PORTA);
    // delay for half secs
    TIMER0_Delay(500);
}

void CarsRedenable(void)
{
    // set cars state to red state
    CarsMode = CARS_RED;
    // turn cars red on
    LED_on(PIN2, PORTA);
}

void CarsRedDisable(void)
{
    // turn cars red off
    LED_off(PIN2, PORTA);
}
```

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense

Description

VA Find References Results Error List

Output

Ready

Type here to search

Solution Explorer

Search Solution Explorer (Ctrl+U)

- application
 - app.c
 - app.h
- led_trafficlayer
 - LED
 - Led.c
 - LED.h
 - TRAFFIC_LIGHTS
 - TRAFFIC_LIGHTS.c
 - TRAFFIC_LIGHTS.h
- middle_layer
 - DIO_DEVICE
 - DIO.c
 - DIO.h
 - EXTERNAL_INTERRUPT
 - EXT_INT.c

Properties

VA View VA Outline Solution Explorer

Project File Line

Activate Windows Go to Settings to activate Windows.

Ln 1 Col 1 Ch 1 INS

15°C 10:15 PM ENG 2/22/2023

Application:

On-demand Traffic light control - Microchip Studio

File Edit View VASSISTX ASF Project Build Debug Tools Window Help

Advanced Mode Quick Launch (Ctrl+Q)

TIMER0.h TIMER0.c TRAFFIC_LIGHTS.h TRAFFIC_LIGHTS.c EXT_INT.h EXT_INT.c app.h EXT_INT.c app.c LED.h Led.c

app.h

```
/*
 * app.h
 * 
 * Created: 22/02/2023
 * Author: osama
 */

#ifndef APP_H_
#define APP_H_
//functions prototypes

void appInit(void);
void appStart(void);

#endif /* APP_H_ */
```

Error List

Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense

Description

VA Find References Results Error List

Output

Ready

Type here to search

Solution Explorer

Search Solution Explorer (Ctrl+U)

- application
 - app.c
 - app.h
- led_trafficlayer
 - LED
 - Led.c
 - LED.h
 - TRAFFIC_LIGHTS
 - TRAFFIC_LIGHTS.c
 - TRAFFIC_LIGHTS.h
- middle_layer
 - DIO_DEVICE
 - DIO.c
 - DIO.h
 - EXTERNAL_INTERRUPT
 - EXT_INT.c

Properties

VA View VA Outline Solution Explorer

Project File Line

Activate Windows Go to Settings to activate Windows.

Ln 1 Col 1 Ch 1 INS

15°C 10:16 PM ENG 2/22/2023

The screenshot shows the Microchip Studio IDE interface. The main window displays the source code for `applint.c`. The code includes comments indicating it was created on 22/02/2023 by Osama. It defines a function `void applint(void)` which contains logic for handling a long press on pin 2. The logic involves reading the value of pin 2, setting a delay of 500ms, and then reading the value again. If the value has changed, it indicates a long press and the program exits. Otherwise, it returns immediately to continue normal mode. The code also includes includes for `LED.h`, `TRAFFIC_LIGHTS.h`, `EXT_INT.h`, and `app.h`. The Solution Explorer on the right shows the project structure with files like `application.c`, `application.h`, `LED.c`, `LED.h`, `TRAFFIC_LIGHTS.c`, `TRAFFIC_LIGHTS.h`, `DIO.c`, `DIO.h`, and `EXT_INT.c`.

```
#include "applint.h"
/* Created: 22/02/2023
 * Author: osama
 */
#include "../led_tracflayer/LED/LED.h"
#include "../middle_layer/TIMER0_NORMAL/TIMER0.h"
#include "../middle_layer/EXTERNAL_INTERRUPT/EXT_INT.h"
#include "../led_tracflayer/TRAFFIC_LIGHTS/TRAFFIC_LIGHTS.h"
#include "app.h"

ISR(INT0_vect)
{
    // delay to detect long press and make no action
    uint8_t value;
    TIMER0_Delay(500);
    DIO_read(PIN2, PORTD, &value);
    if (value == 0)
    {
        // long press detected
        // nothing here
        // return immediately to continue normal mode
    }
    else
}
```