EV Control Task

AUTHOR
Version 0.1
Tue Aug 24 2021

Table of Contents

Table of contents

Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

File Index

File List

Here is a list of all files with brief descriptions:

Data Structure Documentation

GPIO Struct Reference

GPIO PORT Definition using struct of registers.
`#include <GPIO.h>`

Data Fields

Register PIN
Register DDR
Register Port

Detailed Description

GPIO PORT Definition using struct of registers.

Field Documentation

Register DDR

Register PIN

Register Port

The documentation for this struct was generated from the following file:

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/**GPIO.h**

Register Union Reference

Register Definition using unions.
`#include <GPIO.h>`

Data Fields

vu8 AllRegister
struct {
  vu8 Bit0:1
  vu8 Bit1:1
  vu8 Bit2:1
  vu8 Bit3:1
  vu8 Bit4:1
  vu8 Bit5:1
  vu8 Bit6:1
  vu8 Bit7:1
} **Bits**

Detailed Description

Register Definition using unions.

Field Documentation

vu8 AllRegister

vu8 Bit0

vu8 Bit1

vu8 Bit2

vu8 Bit3

vu8 Bit4

vu8 Bit5

vu8 Bit6

vu8 Bit7

struct { ... } Bits

The documentation for this union was generated from the following file:

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/**GPIO.h**

File Documentation

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/GPIO.c File Reference

```
#include "GPIO.h"
#include "Macros.h"
```

Functions

void **SetDataDirection** (**GPIO** *gpio, **Mode** mode, **Pin_Number** pin)
void **Output** (**GPIO** *gpio, **Value** value, **Pin_Number** pin)
void **AllPortOutput** (**GPIO** *gpio)

Function Documentation

void AllPortOutput (GPIO * *gpio*)

Making the entire port pins output

void Output (GPIO * *gpio*, Value *value*, Pin_Number *pin*)

Giving the output pins value 1 or 0

void SetDataDirection (GPIO * *gpio*, Mode *mode*, Pin_Number *pin*)

Making the pin input or output

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/GPIO.h File Reference

```
#include "StandardTypes.h"
```

Data Structures

union **Register**

Register *Definition using unions.*

struct **GPIO**

GPIO *PORT Definition using struct of registers.*

Macros

#define **GPIOA** ((**GPIO***)0x39)
#define **GPIOB** ((**GPIO***)0x36)
#define **GPIOC** ((**GPIO***)0x33)
#define **GPIOD** ((**GPIO***)0x30)

Enumerations

enum **Value** { **LOW**, **HIGH** }
enum **Mode** { **PullUp**, **Floating**, **output** }
enum **Pin_Number** { **PIN0**, **PIN1**, **PIN2**, **PIN3**, **PIN4**, **PIN5**, **PIN6**, **PIN7** }

Functions

void **SetDataDirection** (**GPIO** *gpio, **Mode** mode, **Pin_Number** pin)
void **Output** (**GPIO** *gpio, **Value** value, **Pin_Number** pin)
void **AllPortOutput** (**GPIO** *gpio)

Macro Definition Documentation

#define GPIOA ((GPIO*)0x39)

#define GPIOB ((GPIO*)0x36)

#define GPIOC ((GPIO*)0x33)

#define GPIOD ((GPIO*)0x30)

Enumeration Type Documentation

enum Mode

Enumerator:

PullUp	
Floating	
output	

enum Pin_Number

Enumerator:

PIN0	
PIN1	
PIN2	
PIN3	
PIN4	
PIN5	
PIN6	
PIN7	

enum Value

Enumerator:

LOW	
HIGH	

Function Documentation

void AllPortOutput (GPIO * *gpio*)

Making the entire port pins output

void Output (GPIO * *gpio*, Value *value*, Pin_Number *pin*)

Giving the output pins value 1 or 0

void SetDataDirection (GPIO * *gpio*, Mode *mode*, Pin_Number *pin*)

Making the pin input or output

GPIO.h

```
Go to the documentation of this file.1 /*
2  * GPIO.h
3  *
4  * Created: 2021-08-18 10:49:48 AM
5  *  Author: Abdelaziz Mohammad
6  */
7
8
9 #ifndef GPIO_H
10 #define GPIO_H
11
12 #include "StandardTypes.h"
13
14 #define GPIOA  ((GPIO*)0x39)
15 #define GPIOB  ((GPIO*)0x36)
16 #define GPIOC  ((GPIO*)0x33)
17 #define GPIOD  ((GPIO*)0x30)
18
20 typedef union
21 {
22
23
24     vu8 AllRegister;
25     struct
26     {
27         vu8 Bit0:1 ;
28         vu8 Bit1:1 ;
29         vu8 Bit2:1 ;
30         vu8 Bit3:1 ;
31         vu8 Bit4:1 ;
32         vu8 Bit5:1 ;
33         vu8 Bit6:1 ;
34         vu8 Bit7:1 ;
35     }Bits;
36
37 }Register;
38
40 typedef struct
41 {
42     Register PIN;
43     Register DDR;
44     Register Port;
45
46 }GPIO;
47
48 typedef enum
49 {
50     LOW,
51     HIGH,
52
53 }Value;
54
55 typedef enum
56 {
57     PullUp,
58     Floating,
59     output,
60
61 }Mode;
62
63
64
65 typedef enum
66 {
67     PIN0,
68     PIN1,
69     PIN2,
70     PIN3,
71     PIN4,
72     PIN5,
73     PIN6,
74     PIN7,
75
```

```
76 }Pin_Number;
77
78
79
80
81
82
83 void SetDataDirection (GPIO *gpio , Mode mode , Pin Number pin );
84 void Output (GPIO *gpio , Value value , Pin Number pin);
85 void  AllPortOutput(GPIO* gpio);
86
87 #endif /* GPIO_H_ */
```

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/LCD.c File Reference

```
#include "LCD.h"
#include <util/delay.h>
```

Macros

#define **F_CPU** 8000000

Functions

void **LCD_Init** ()
LCD initialization Function Prototype.

void **LCD_WriteChar** (**u8** Char)
LCD WriteChar Function Prototype.

void **LCD_WriteSentence** (const **u8** *Sentence)
LCD WriteSentence Function Prototype.

void **LCD_Clear** ()
LCD Clear Function Prototype.

void **LCD_MoveCursor** (**u8** x, **u8** y)
LCD WriteChar Function Prototype.

Macro Definition Documentation

#define F_CPU 8000000

Function Documentation

void LCD_Clear (void)

LCD Clear Function Prototype.

void LCD_Init (void)

LCD initialization Function Prototype.

Delay said in DataSheet
Initialization of RS,RW,E
Initialization of DataPort as Output
Send Command to clear Display
Function Set part

Display Part
Entry Part

void LCD_MoveCursor (u8 *x*, u8 *y*)

LCD WriteChar Function Prototype.

x represents the coloumn and y represents the row

void LCD_WriteChar (u8 *Char*)

LCD WriteChar Function Prototype.

First For Data **Register** RS=1
Second For write RW=0
Third Set E to high
Forth PUT command in LCD_Data_Port
Fifth Clear E

void LCD_WriteSentence (const u8 * *Sentence*)

LCD WriteSentence Function Prototype.

Printing

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/LCD.h File Reference

```
#include "GPIO.h"
#include "StandardTypes.h"
```

Macros

#define **LCD_Control_Port** **GPIOA**
LCD Control PORT.

#define **RS** **PIN5**
#define **RW** **PIN6**
#define **E** **PIN7**
#define **LCD_Data_Port** **GPIOB**
LCD DATA PORT.

#define **Return_Home** 0x02
LCD Config.

#define **Clear_Display** 0x01
#define **_8bit_2Lines_5x8Font** 0x38
#define **DisplayOn_CursorOff_BlinkCursorOff** 0x0C
#define **CursorMovesRight** 0x06
#define **_4bit_2Lines_5x8Font** 0x28

Functions

void **LCD_Init** (void)
LCD initialization Function Prototype.

void **LCD_WriteChar** (**u8** Char)
LCD WriteChar Function Prototype.

void **LCD_MoveCursor** (**u8** x, **u8** y)
LCD WriteChar Function Prototype.

void **LCD_WriteSentence** (const **u8** *Sentence)
LCD WriteSentence Function Prototype.

void **LCD_Clear** (void)
LCD Clear Function Prototype.

Macro Definition Documentation

#define _4bit_2Lines_5x8Font 0x28

#define _8bit_2Lines_5x8Font 0x38

#define Clear_Display 0x01

#define CursorMovesRight 0x06

#define DisplayOn_CursorOff_BlinkCursorOff 0x0C

#define E PIN7

#define LCD_Control_Port GPIOA

LCD Control PORT.

#define LCD_Data_Port GPIOB

LCD DATA PORT.

#define Return_Home 0x02

LCD Config.

#define RS PIN5

#define RW PIN6

Function Documentation

void LCD_Clear (void)

LCD Clear Function Prototype.

void LCD_Init (void)

LCD initialization Function Prototype.

Delay said in DataSheet
Initialization of RS,RW,E
Initialization of DataPort as Output
Send Command to clear Display
Function Set part
Display Part
Entry Part

void LCD_MoveCursor (u8 *x*, u8 *y*)

LCD WriteChar Function Prototype.

x represents the coloumn and y represents the row

void LCD_WriteChar (u8 *Char*)

LCD WriteChar Function Prototype.

First For Data **Register** RS=1
Second For write RW=0
Third Set E to high
Forth PUT command in LCD_Data_Port
Fifth Clear E

void LCD_WriteSentence (const u8 * *Sentence*)

LCD WriteSentence Function Prototype.

Printing

LCD.h

Go to the documentation of this file.

```
1  /*
2   * LCD.h
3   *
4   * Created: 2021-08-22 11:49:33 AM
5   *  Author: Abdelaziz Mohammad
6   */
7
8
9  #ifndef LCD H
10 #define LCD H
11
12
13 #include "GPIO.h"
14 #include "StandardTypes.h"
15
16
17
18
19
21
22 #define     LCD Control Port    GPIOA
23 #define     RS          PIN5
24 #define     RW          PIN6
25 #define     E           PIN7
27 #define     LCD_Data_Port       GPIOB
29 #define     Return Home                         0x02
30 #define     Clear Display                       0x01
31 #define     _8bit_2Lines_5x8Font                0x38
32 #define     DisplayOn_CursorOff_BlinkCursorOff  0x0C
33 #define     CursorMovesRight                    0x06
34 #define      4bit 2Lines 5x8Font                0x28
35
37 void LCD_Init(void);
39 void LCD_WriteChar(u8 Char);
41 void LCD MoveCursor(u8 x,u8 y);
43 void LCD WriteSentence(const u8* Sentence);
45 void LCD Clear(void);
46
47
48
49 #endif /* LCD_H_ */
```

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/Macros.h File Reference

Macros

#define **SetBit**(Reg, Bit) Reg |= (1<<Bit)

Macros for Setting, Clearing, Toggling and Checking Bit.

#define **ClearBit**(Reg, Bit) Reg &= ~(1<<Bit)
#define **ToggleBit**(Reg, Bit) Reg ^= (1<<Bit)
#define **CheckBit**(Reg, Bit) ((Reg>>Bit) & 1)

Macro Definition Documentation

#define CheckBit(Reg, Bit) ((Reg>>Bit) & 1)

#define ClearBit(Reg, Bit) Reg &= ~(1<<Bit)

#define SetBit(Reg, Bit) Reg |= (1<<Bit)

Macros for Setting, Clearing, Toggling and Checking Bit.

#define ToggleBit(Reg, Bit) Reg ^= (1<<Bit)

Macros.h

Go to the documentation of this file.

```
1 /*
2  * Macros.h
3  *
4  * Created: 2021-08-18 10:44:55 AM
5  *  Author: sigmaa
6  */
7
8
9 #ifndef MACROS H
10 #define MACROS H
11
13 #define SetBit(Reg,Bit) Reg |= (1<<Bit)
14 #define ClearBit(Reg,Bit) Reg &= ~(1<<Bit)
15 #define ToggleBit(Reg,Bit) Reg ^= (1<<Bit)
16 #define CheckBit(Reg,Bit)  ((Reg>>Bit) & 1)
17
18 #endif /* MACROS_H_ */
```

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/main.c File Reference

```
#include "StandardTypes.h"
#include "Macros.h"
#include "GPIO.h"
#include "LCD.h"
#include "UART_Tx.h"
#include "Timer1.h"
#include <avr/interrupt.h>
```

Functions

int **main** (void)
ISR (TIMER1_COMPA_vect)

Variables

u8 delay_Flag = 1
u8 Flag_Tx = 2
Delay Flag for 3 seconds delay.

u8 Flag_Rx = '.'
u8 Flag_Button_Click =1

Function Documentation

ISR (TIMER1_COMPA_vect)

int main (void)

Program Entry point

Variable Documentation

u8 delay_Flag = 1

u8 Flag_Button_Click =1

u8 Flag_Rx = '.'

u8 Flag_Tx = 2

Delay Flag for 3 seconds delay.

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/StandardTypes.h File Reference

Typedefs

typedef unsigned char **u8**
typedef unsigned short **u16**
typedef unsigned long long **u64**
typedef volatile unsigned char **vu8**
typedef volatile unsigned short **vu16**

Typedef Documentation

typedef unsigned short u16

typedef unsigned long long u64

typedef unsigned char u8

typedef volatile unsigned short vu16

typedef volatile unsigned char vu8

StandardTypes.h

Go to the documentation of this file.

```
1 /*
2  * StandardTypes.h
3  *
4  * Created: 2021-08-18 10:47:13 AM
5  *  Author: sigmaa
6  */
7
8
9 #ifndef STANDARDTYPES H
10 #define STANDARDTYPES H
11
12 typedef unsigned char u8;
13 typedef unsigned short u16;
14 typedef  unsigned long long u64;
15 typedef volatile unsigned char vu8;
16 typedef volatile unsigned short vu16;
17
18
19
20 #endif /* STANDARDTYPES_H_ */
```

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/Timer1.c File Reference

```
#include "Timer1.h"
#include "GPIO.h"
#include "Macros.h"
#include "StandardTypes.h"
```

Functions

void **EnableGlobalInterrupt** ()
void **delay_Init** ()
void **timer_stop** ()
void **timer_start** ()
void **delay_3_Seconds** ()
void **EnableInterruptTimer1** ()
void **DisableInterruptTimer1** ()

Function Documentation

void delay_3_Seconds ()

Put number 23438 in OCR

void delay_Init ()

CTC with ICU TOP
CTC Set on compare
1024 prescaler

void DisableInterruptTimer1 ()

Clearing PIN4 in TIMSK **Register**

void EnableGlobalInterrupt ()

void EnableInterruptTimer1 ()

Setting PIN4 in TIMSK **Register**

void timer_start ()

void timer_stop ()

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/Timer1.h File Reference

```
#include "GPIO.h"
#include "StandardTypes.h"
```

Macros

#define **TCCR1A** ((**Register***)0x4F)
#define **TCCR1B** ((**Register***)0x4E)
#define **OCR1AH** ((**Register***)0x4B)
#define **OCR1AL** (*((volatile unsigned short*)0x4A))
#define **OCR1BH** ((**Register***)0x49)
#define **OCR1BL** ((**Register***)0x48)
#define **TIMSK** ((**Register***)0x59)
#define **ICUH** ((**Register***)0x47)
#define **ICUL** (*((volatile unsigned short*)0x46))
#define **SREG** ((**Register***)0x5F)
#define **TCNT** (*((volatile unsigned short*)0x4C))
#define **WGM10** 0
#define **WGM11** 1
#define **WGM12** 3
#define **WGM13** 4
#define **COM1A0** 6
#define **COM1A1** 7
#define **COM1B0** 4
#define **COM1B1** 5
#define **CS10** 0
#define **CS11** 1
#define **CS12** 2
#define **TOIE1** 2

Functions

void **EnableGlobalInterrupt** ()
void **delay_Init** ()
void **delay_3_Seconds** ()
void **EnableInterruptTimer1** ()
void **DisableInterruptTimer1** ()
void **timer_stop** ()
void **timer_start** ()

Macro Definition Documentation

#define COM1A0 6

#define COM1A1 7

#define COM1B0 4

#define COM1B1 5

#define CS10 0

#define CS11 1

#define CS12 2

#define ICUH ((Register*)0x47)

#define ICUL (*((volatile unsigned short*)0x46))

#define OCR1AH ((Register*)0x4B)

#define OCR1AL (*((volatile unsigned short*)0x4A))

#define OCR1BH ((Register*)0x49)

#define OCR1BL ((Register*)0x48)

#define SREG ((Register*)0x5F)

#define TCCR1A ((Register*)0x4F)

#define TCCR1B ((Register*)0x4E)

#define TCNT (*((volatile unsigned short*)0x4C))

#define TIMSK ((Register*)0x59)

#define TOIE1 2

#define WGM10 0

#define WGM11 1

#define WGM12 3

#define WGM13 4

Function Documentation

void delay_3_Seconds ()

Put number 23438 in OCR

void delay_Init ()

CTC with ICU TOP
CTC Set on compare
1024 prescaler

void DisableInterruptTimer1 ()

Clearing PIN4 in TIMSK **Register**

void EnableGlobalInterrupt ()

void EnableInterruptTimer1 ()

Setting PIN4 in TIMSK **Register**

void timer_start ()

void timer_stop ()

Timer1.h

```
Go to the documentation of this file.1 /*
2  * Timer1.h
3  *
4  * Created: 2021-08-21 6:01:55 AM
5  *  Author: sigmaa
6  */
7
8
9 #ifndef TIMER1_H
10 #define TIMER1_H
11
12 #include  "GPIO.h"
13 #include  "StandardTypes.h"
14
15 #define  TCCR1A     ((Register*)0x4F)
16 #define  TCCR1B     ((Register*)0x4E)
17 #define  OCR1AH     ((Register*)0x4B)
18 #define  OCR1AL     (*((volatile unsigned short*)0x4A))
19 #define  OCR1BH     ((Register*)0x49)
20 #define  OCR1BL     ((Register*)0x48)
21 #define  TIMSK      ((Register*)0x59)
22 #define  ICUH       ((Register*)0x47)
23 #define  ICUL       (*((volatile unsigned short*)0x46))
24 #define  SREG       ((Register*)0x5F)
25 #define  TCNT       (*((volatile unsigned short*)0x4C))
26
27
28 #define WGM10  0
29 #define WGM11  1
30 #define WGM12  3
31 #define WGM13  4
32 #define COM1A0 6
33 #define COM1A1 7
34 #define COM1B0 4
35 #define COM1B1 5
36 #define CS10   0
37 #define CS11   1
38 #define CS12   2
39 #define TOIE1 2
40
41 void EnableGlobalInterrupt();
42 void delay Init();
43 void delay 3 Seconds();
44 void EnableInterruptTimer1();
45 void DisableInterruptTimer1();
46 void timer_stop();
47 void timer_start();
48 #endif /* TIMER1_H_ */
```

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/UART_Tx.c File Reference

```
#include "UART_Tx.h"
#include "Macros.h"
```

Functions

void **UART_init** (**UART_Mode** mode)
UART initialization Function Prototype.

void **UART_SendChar** (**u8** Data)
UART SendChar Function Prototype.

u8 UART_ReceiveChar (void)
UART ReceiveChar Function Prototype.

Function Documentation

void UART_init (UART_Mode *mode*)

UART initialization Function Prototype.

Input pin for Rx
Output pin for Tx
Option to activate Trasmitter Mode
Option to Receiver Mode
No Parity
1 StopBit
8_Bits
For 9600 Baud Rate and 8MHz Freq

u8 UART_ReceiveChar (void)

UART ReceiveChar Function Prototype.

When RXC =1 that means there is data in UDR

void UART_SendChar (u8 *Data*)

UART SendChar Function Prototype.

When UDRE = 1 that means the register is empty and data is in shift register ready to be sent.

C:/Users/sigmaa/Desktop/Abdelaziz Mohammad - EV control task/Task/Task/UART_Tx.h File Reference

```
#include "GPIO.h"
#include "StandardTypes.h"
```

Macros

#define **UDR** ((**Register***)0x2C)
#define **UCSRA** ((**Register***)0x2B)
#define **UCSRB** ((**Register***)0x2A)
#define **UCSRC** ((**Register***)0x40)
#define **UBRRL** ((**Register***)0x29)
#define **UBRRH** ((**Register***)0x40)
#define **RXC** 7 /*Receive Complete*/
#define **TXC** 6 /*Transmit Complete*/
#define **UDRE** 5 /*Data **Register** Empty*/
#define **RXEN** 4 /*Receive Enable*/
#define **TXEN** 3 /*Transmit Enable*/
#define **UMP0** 4
#define **UMP1** 5
#define **UCSZ0** 1
#define **UCSZ1** 2
#define **UCSZ2** 2
#define **USBS** 3

Enumerations

enum **UART_Mode** { **Transmitter**, **Receiver** }

Functions

void **UART_init** (**UART_Mode** mode)
UART initialization Function Prototype.

void **UART_SendChar** (**u8** Data)
UART SendChar Function Prototype.

u8 UART_ReceiveChar (void)
UART ReceiveChar Function Prototype.

Macro Definition Documentation

#define RXC 7 /*Receive Complete*/

#define RXEN 4 /*Receive Enable*/

#define TXC 6 /*Transmit Complete*/

#define TXEN 3 /*Transmit Enable*/

#define UBRRH ((Register*)0x40)

#define UBRRL ((Register*)0x29)

#define UCSRA ((Register*)0x2B)

#define UCSRB ((Register*)0x2A)

#define UCSRC ((Register*)0x40)

#define UCSZ0 1

#define UCSZ1 2

#define UCSZ2 2

#define UDR ((Register*)0x2C)

#define UDRE 5 /*Data Register Empty*/

#define UMP0 4

#define UMP1 5

#define USBS 3

Enumeration Type Documentation

enum UART_Mode

Enumerator:

Transmitter	
Receiver	

Function Documentation

void UART_init (UART_Mode *mode*)

UART initialization Function Prototype.

Input pin for Rx

Output pin for Tx

Option to activate Trasmitter Mode

Option to Receiver Mode

No Parity

1 StopBit

8_Bits

For 9600 Baud Rate and 8MHz Freq

u8 UART_ReceiveChar (void)

UART ReceiveChar Function Prototype.

When RXC =1 that means there is data in UDR

void UART_SendChar (u8 *Data*)

UART SendChar Function Prototype.

When UDRE = 1 that means the register is empty and data is in shift register ready to be sent.

UART_Tx.h

Go to the documentation of this file.

```
1  /*
2   * UART_Tx.h
3   *
4   * Created: 2021-08-20 4:21:46 AM
5   *  Author: sigmaa
6   */
7
8
9  #ifndef UART TX H
10 #define UART TX H
11
12
13 #include "GPIO.h"
14 #include "StandardTypes.h"
15
16 #define UDR   ((Register*)0x2C)
17 #define UCSRA ((Register*)0x2B)
18 #define UCSRB ((Register*)0x2A)
19 #define UCSRC ((Register*)0x40)
20 #define UBRRL ((Register*)0x29)
21 #define UBRRH ((Register*)0x40)
22
23
24 /* Special Bits */
25
26 #define RXC    7   /*Receive  Complete*/
27 #define TXC    6   /*Transmit Complete*/
28 #define UDRE   5   /*Data Register Empty*/
29 #define RXEN   4   /*Receive Enable*/
30 #define TXEN   3   /*Transmit Enable*/
31 #define UMP0        4
32 #define UMP1        5
33 #define UCSZ0       1
34 #define UCSZ1       2
35 #define UCSZ2       2
36 #define USBS        3
37
38
39 typedef enum
40 {
41
42     Transmitter,
43     Receiver,
44 }UART Mode;
45
47 void UART_init(UART_Mode mode);
49 void UART_SendChar(u8 Data);
51 u8 UART ReceiveChar(void);
52
53
54
55
56 #endif /* UART_TX_H_ */
```

Index

INDEX