

SWC_STEPPER

Version v1.0
10/16/2023 5:30:00 PM

Table of Contents

Data Structure Index.....	2
File Index.....	3
Data Structure Documentation	4
LBTY_tuniPort16.....	4
LBTY_tuniPort8.....	6
File Documentation	8
main.c	8
STEPPER_cfg.h	9
STEPPER_int.h	12
STEPPER_prg.c	15
STEPPER_priv.h	17
H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/LBIT_int.h	19
H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/LBIT_int.h	22
H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/LBTY_int.h	24
H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/LBTY_int.h	29
H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/LCTY_int.h	32
H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/LCTY_int.h	33
Index.....	Error! Bookmark not defined.

Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

<u>LBTY_tuniPort16</u>4
<u>LBTY_tuniPort8</u>6

File Index

File List

Here is a list of all files with brief descriptions:

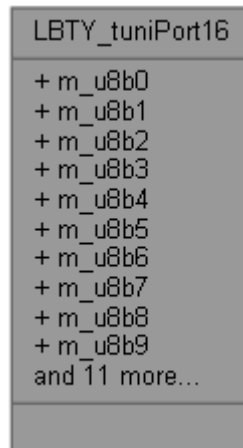
main.c	8
STEPPER_cfg.h	9
STEPPER_int.h	12
STEPPER_prg.c	15
STEPPER_priv.h	17
H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/ LBIT_int.h	19
H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/ LBTY_int.h	24
H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/ LCTY_int.h	32

Data Structure Documentation

LBTY_tuniPort16 Union Reference

```
#include <LBTY_int.h>
```

Collaboration diagram for LBTY_tuniPort16:



Data Fields

- struct {
 - [u8 m_u8b0](#):1
 - [u8 m_u8b1](#):1
 - [u8 m_u8b2](#):1
 - [u8 m_u8b3](#):1
 - [u8 m_u8b4](#):1
 - [u8 m_u8b5](#):1
 - [u8 m_u8b6](#):1
 - [u8 m_u8b7](#):1
 - [u8 m_u8b8](#):1
 - [u8 m_u8b9](#):1
 - [u8 m_u8b10](#):1
 - [u8 m_u8b11](#):1
 - [u8 m_u8b12](#):1
 - [u8 m_u8b13](#):1
 - [u8 m_u8b14](#):1
 - [u8 m_u8b15](#):1
 - } [sBits](#)
 - struct {
 - [u8 m_u8low](#)
 - [u8 m_u8high](#)
 - } [sBytes](#)
 - [u16 u_u16Word](#)
-

Field Documentation

[u8](#) m_u8b0

[u8](#) m_u8b1

[u8](#) m_u8b10

[u8](#) m_u8b11

[u8](#) m_u8b12

[u8](#) m_u8b13

[u8](#) m_u8b14

[u8](#) m_u8b15

[u8](#) m_u8b2

[u8](#) m_u8b3

[u8](#) m_u8b4

[u8](#) m_u8b5

[u8](#) m_u8b6

[u8](#) m_u8b7

[u8](#) m_u8b8

[u8](#) m_u8b9

[u8](#) m_u8high

[u8](#) m_u8low

struct { ... } sBits

struct { ... } sBytes

[u16](#) u_u16Word

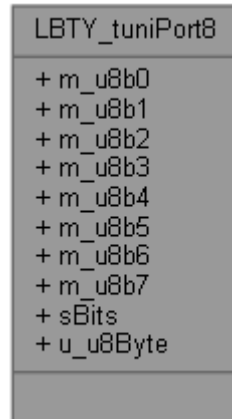
The documentation for this union was generated from the following file:

- H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/[LBTY_int.h](#)

LBTY_tuniPort8 Union Reference

```
#include <LBTY_int.h>
```

Collaboration diagram for LBTY_tuniPort8:



Data Fields

- struct {
- [u8 m_u8b0](#):1
- [u8 m_u8b1](#):1
- [u8 m_u8b2](#):1
- [u8 m_u8b3](#):1
- [u8 m_u8b4](#):1
- [u8 m_u8b5](#):1
- [u8 m_u8b6](#):1
- [u8 m_u8b7](#):1
- } [sBits](#)
- [u8 u_u8Byte](#)

Detailed Description

Union Byte bit by bit

Field Documentation

[u8](#) m_u8b0

[u8](#) m_u8b1

[u8](#) m_u8b2

[u8](#) m_u8b3

[u8](#) m_u8b4

[u8](#) m_u8b5

[u8](#) m_u8b6

[u8](#) m_u8b7

struct { ... } sBits

[u8](#) u_u8Byte

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

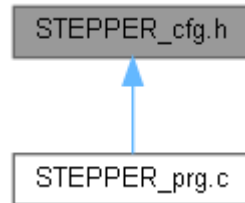
- H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/[LBTY_int.h](#)

File Documentation

main.c File Reference

STEPPER_cfg.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define [STEPPER_COIL_STEP](#) [STEPPER_Full_Step](#)
 - #define [STEPPER_COIL_NORMAL](#) PIN_Low
 - #define [STEPPER_COIL_MAX](#) 4u
 - #define [STEPPER_PORT_COIL0](#) C
 - #define [STEPPER_PIN_COIL0](#) 0u
 - #define [STEPPER_PORT_COIL1](#) C
 - #define [STEPPER_PIN_COIL1](#) 1u
 - #define [STEPPER_PORT_COIL2](#) C
 - #define [STEPPER_PIN_COIL2](#) 2u
 - #define [STEPPER_PORT_COIL3](#) C
 - #define [STEPPER_PIN_COIL3](#) 3u
 - #define [STEPPER_STEP_ANGLE](#) 10.0f
 - #define [STEPPER_MIN_STEP_DELAY](#) 125u
 - #define [STEPPER_NUM_DELAY](#) 130u
-

Macro Definition Documentation

```
#define STEPPER_COIL_MAX 4u

#define STEPPER_COIL_NORMAL PIN_Low

#define STEPPER_COIL_STEP STEPPER\_Full\_Step

#define STEPPER_MIN_STEP_DELAY 125u

#define STEPPER_NUM_DELAY 130u

#define STEPPER_PIN_COIL0 0u

#define STEPPER_PIN_COIL1 1u

#define STEPPER_PIN_COIL2 2u

#define STEPPER_PIN_COIL3 3u

#define STEPPER_PORT_COIL0 C

#define STEPPER_PORT_COIL1 C

#define STEPPER_PORT_COIL2 C

#define STEPPER_PORT_COIL3 C

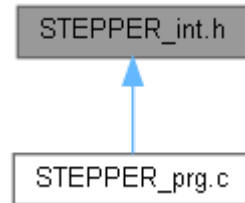
#define STEPPER_STEP_ANGLE 10.0f
```

STEPPER_cfg.h

```
Go to the documentation of this file.1 /*
*****
2 /* ***** FILE DEFINITION SECTION ***** */
3 /* ***** */
4 /* File Name      : STEPPER_cfg.h */
5 /* Author         : MAAM */
6 /* Version        : v01.2 */
7 /* date           : Mar 25, 2023 */
8 /* ***** */
9 /* ***** HEADER FILES INCLUDES ***** */
10 /* ***** */
11
12 #ifndef STEPPER_CFG_H_
13 #define STEPPER_CFG_H_
14
15 /* ***** */
16 /* ***** TYPE_DEF/STRUCT/ENUM SECTION ***** */
17 /* ***** */
18
19 /* ***** */
20 /* ***** MACRO/DEFINE SECTION ***** */
21 /* ***** */
22
23 #define STEPPER_COIL_STEP      STEPPER_Full_Step
24 #define STEPPER_COIL_NORMAL    PIN_Low
25 #define STEPPER_COIL_MAX       4u
26
27 #define STEPPER_PORT_COIL0      C
28 #define STEPPER_PIN_COIL0       0u
29
30 #define STEPPER_PORT_COIL1      C
31 #define STEPPER_PIN_COIL1       1u
32
33 #define STEPPER_PORT_COIL2      C
34 #define STEPPER_PIN_COIL2       2u
35
36 #define STEPPER_PORT_COIL3      C
37 #define STEPPER_PIN_COIL3       3u
38
39 #define STEPPER_STEP_ANGLE      10.0f    //1.8f
40 #define STEPPER_MIN_STEP_DELAY  125u
41 #define STEPPER_NUM_DELAY       130u
42
43 /* ***** */
44 /* ***** CONST SECTION ***** */
45 /* ***** */
46
47 /* ***** */
48 /* ***** VARIABLE SECTION ***** */
49 /* ***** */
50
51 /* ***** */
52 /* ***** FUNCTION SECTION ***** */
53 /* ***** */
54
55
56 #endif /* STEPPER_CFG_H_ */
57 /***** E N D (STEPPER_cfg.h) *****/
```

STEPPER_int.h File Reference

This graph shows which files directly or indirectly include this file:



Enumerations

- enum [STEPPER_tenuDirection](#) { [STEPPER_WithClockWise](#) = (u8)0u, [STEPPER_OppositClockWise](#), [STEPPER_Stop](#) }

Functions

- void [STEPPER_vidInit](#) (void)
- void [STEPPER_vidCycle](#) (u16 u16DelayValue, u8 u8Direction)
- void [STEPPER_vidStep](#) (u16 u16DelayValue, u8 u8Direction)

Enumeration Type Documentation

enum [STEPPER_tenuDirection](#)

Enumerator:

STEPPER_WithClockWise	
STEPPER_OppositClockWise	
STEPPER_Stop	

```
19 {
20     STEPPER_WithClockWise = (u8)0u,
21     STEPPER_OppositClockWise,
22     STEPPER_Stop
23 } STEPPER_tenuDirection;
```

Function Documentation

void [STEPPER_vidCycle](#) (u16 u16DelayValue, u8 u8Direction)

```
72 {
73     switch (u8Direction){
74         case STEPPER_WithClockWise :
75         case STEPPER_OppositClockWise :
76             for(u16 i = (360u / STEPPER_STEP_ANGLE) ; i-- ; ){
77                 STEPPER_vidStep(u16DelayValue, u8Direction);
78             }
79             break;
80         case STEPPER_Stop :
81             default:
82                 return;
83     }
84 }
```

Here is the call graph for this function:



void STEPPER_vidInit (void)

```
59     {
60         for(u8 i = STEPPER_NUM ; i-- ; ){
61             GPIO_tstrPinConfig* pTemp =
62             (GPIO_tstrPinConfig*)&kau8STEPPERConfiguration_LGB[i];
63             GPIO_u8PinInit(*pTemp);
64             GPIO_u8SetPinValue(pTemp->m_Port, pTemp->m_Pin, pTemp->m_Value);
65         }
```

void STEPPER_vidStep (u16 u16DelayValue, u8 u8Direction)

```
91     {
92         static u8 u8Coil;
93
94         switch (u8Direction){
95             case STEPPER_WithClockWise :
96                 u8Coil = (u8Coil + 1) % ku8StepperNum;
97                 break;
98             case STEPPER_OppositClockWise :
99                 u8Coil = u8Coil ? (u8Coil - 1) : (ku8StepperNum - 1);
100                 break;
101             case STEPPER_Stop :
102                 default:
103                     u8Coil = LBTY u8ZERO;
104                     return;
105         }
106
107         u8 u8Temp = kau8StepperStep[u8Coil];
108         GPIO_tstrPinConfig* pTemp =
109         (GPIO_tstrPinConfig*)&kau8STEPPERConfiguration_LGB;
110         for(u8 i = STEPPER_NUM ; i-- ; pTemp++){
111             GPIO_u8SetPinValue(pTemp->m_Port, pTemp->m_Pin, ((u8Temp & (1<<i)) ?
112             STEPPER_COIL_Set : STEPPER_COIL_Reset));
113         }
114         vidMyDelay_ms(u16DelayValue);
115     }
```

Here is the caller graph for this function:



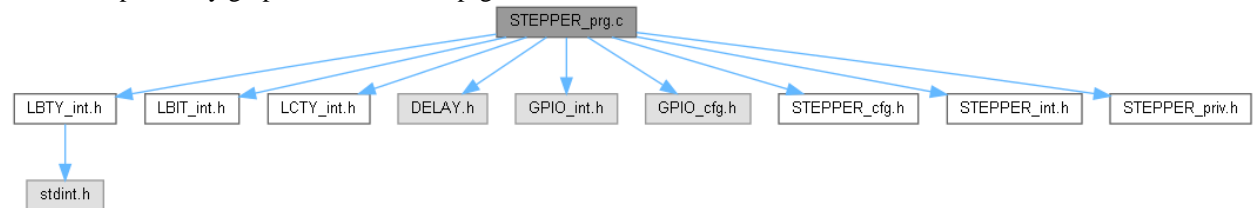
STEPPER_int.h

```
Go to the documentation of this file.1 /*
*****
2 /* ***** FILE DEFINITION SECTION ***** */
3 /* ***** */
4 /* File Name : STEPPER_int.h */
5 /* Author : MAAM */
6 /* Version : v01.2 */
7 /* date : Mar 25, 2023 */
8 /* ***** */
9 /* ***** HEADER FILES INCLUDES ***** */
10 /* ***** */
11
12 #ifndef STEPPER_INT_H_
13 #define STEPPER_INT_H_
14
15 /* ***** */
16 /* ***** TYPE_DEF/STRUCT/ENUM SECTION ***** */
17 /* ***** */
18
19 typedef enum{
20     STEPPER_WithClockWise = (u8)0u,
21     STEPPER_OppositClockWise,
22     STEPPER_Stop
23 }STEPPER_tenuDirection;
24
25 /* ***** */
26 /* ***** MACRO/DEFINE SECTION ***** */
27 /* ***** */
28
29 /* ***** */
30 /* ***** CONST SECTION ***** */
31 /* ***** */
32
33 /* ***** */
34 /* ***** VARIABLE SECTION ***** */
35 /* ***** */
36
37 /* ***** */
38 /* ***** FUNCTION SECTION ***** */
39 /* ***** */
40
41 /* ***** */
42 /* Description : Stepper initialization */
43 /* Input : void */
44 /* Return : void */
45 /* ***** */
46 extern void STEPPER_vidInit(void);
47
48 /* ***** */
49 /* Description : Stepper Run One Cycle */
50 /* Input : u16DelayValue, u8Direction */
51 /* Return : void */
52 /* ***** */
53 extern void STEPPER_vidCycle(u16 u16DelayValue, u8 u8Direction);
54
55 /* ***** */
56 /* Description : Stepper Run Value */
57 /* Input : u16DelayValue, u8Direction */
58 /* Return : void */
59 /* ***** */
60 extern void STEPPER_vidStep(u16 u16DelayValue, u8 u8Direction);
61
62
63 #endif /* STEPPER_INT_H_ */
64 /***** E N D (STEPPER_int.h) *****/
```

STEPPER_prg.c File Reference

```
#include "LBTY_int.h"
#include "LBIT_int.h"
#include "LCTY_int.h"
#include "DELAY.h"
#include "GPIO_int.h"
#include "GPIO_cfg.h"
#include "STEPPER_cfg.h"
#include "STEPPER_int.h"
#include "STEPPER_priv.h"
```

Include dependency graph for STEPPER_prg.c:



Functions

- void [STEPPER_vidInit](#) (void)
- void [STEPPER_vidCycle](#) (u16 u16DelayValue, u8 u8Direction)
- void [STEPPER_vidStep](#) (u16 u16DelayValue, u8 u8Direction)

Variables

- const GPIO_tstrPinConfig [kau8STEPPERConfiguration_LGB](#) [[STEPPER_NUM](#)]
- const u8 [kau8StepperStep](#) [[STEPPER_NUM](#) *2u]
- const u8 [ku8StepperNum](#)

Function Documentation

void STEPPER_vidCycle (u16 u16DelayValue, u8 u8Direction)

```
72                                     {
73     switch (u8Direction){
74         case STEPPER_WithClockWise :
75         case STEPPER_OppositClockWise :
76             for(u16 i = (360u / STEPPER_STEP_ANGLE) ; i-- ; ){
77                 STEPPER_vidStep(u16DelayValue, u8Direction);
78             }
79             break;
80         case STEPPER_Stop :
81         default:
82             return;
83     }
84 }
```

Here is the call graph for this function:



void STEPPER_vidInit (void)

```
59     {
60     for(u8 i = STEPPER_NUM ; i-- ; ){
61         GPIO_tstrPinConfig* pTemp =
62         (GPIO_tstrPinConfig*)&kau8STEPPERConfiguration_LGB[i];
63         GPIO_u8PinInit(*pTemp);
64         GPIO_u8SetPinValue(pTemp->m_Port, pTemp->m_Pin, pTemp->m_Value);
```

```

64     }
65 }

```

void STEPPER_vidStep ([u16](#) u16DelayValue, [u8](#) u8Direction)

```

91     {
92         static u8 u8Coil;
93
94         switch (u8Direction){
95             case STEPPER\_WithClockWise :
96                 u8Coil = (u8Coil + 1) % ku8StepperNum;
97                 break;
98             case STEPPER\_OppositClockWise :
99                 u8Coil = u8Coil ? (u8Coil - 1) : (ku8StepperNum - 1);
100                break;
101             case STEPPER\_Stop :
102             default:
103                 u8Coil = LBTY\_u8ZERO;
104                 return;
105         }
106
107         u8 u8Temp = kau8StepperStep[u8Coil];
108         GPIO_tstrPinConfig* pTemp =
109         (GPIO_tstrPinConfig*)kau8STEPPERConfiguration\_LGB;
110         for(u8 i = STEPPER\_NUM ; i-- ; pTemp++){
111             GPIO_u8SetPinValue(pTemp->m_Port, pTemp->m_Pin, ((u8Temp & (1<<i)) ?
112             STEPPER\_COIL\_Set : STEPPER\_COIL\_Reset));
113         }
114         vidMyDelay_ms(u16DelayValue);
115     }

```

Here is the caller graph for this function:



Variable Documentation

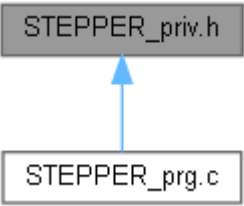
const [GPIO_tstrPinConfig](#)
[kau8STEPPERConfiguration_LGB](#)[[STEPPER_NUM](#)][[extern](#)]

const [u8](#) [kau8StepperStep](#)[[STEPPER_NUM](#)*2u][[extern](#)]

const [u8](#) [ku8StepperNum](#) [[extern](#)]

STEPPER_priv.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define [STEPPER_NUM](#) 4u
- #define [STEPPER_Half_Step](#) 0u
- #define [STEPPER_Full_Step](#) 1u

Enumerations

- enum [STEPPER_tenuState](#) { [STEPPER_COIL_Reset](#) = STEPPER_COIL_NORMAL, [STEPPER_COIL_Set](#) = !STEPPER_COIL_Reset }

Macro Definition Documentation

#define STEPPER_Full_Step 1u

#define STEPPER_Half_Step 0u

#define STEPPER_NUM 4u

Enumeration Type Documentation

enum [STEPPER_tenuState](#)

Enumerator:

STEPPER_COIL_Reset	
STEPPER_COIL_Set	

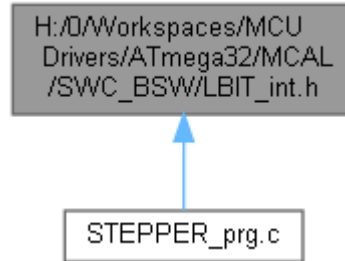
```
19 {
20     STEPPER\_COIL\_Reset = STEPPER\_COIL\_NORMAL,
21     STEPPER\_COIL\_Set = !STEPPER\_COIL\_Reset
22 } STEPPER\_tenuState;
```

STEPPER_priv.h

```
Go to the documentation of this file.1 /*
*****
2 /* ***** FILE DEFINITION SECTION ***** */
3 /* ***** */
4 /* File Name      : STEPPER_priv.h */
5 /* Author         : MAAM */
6 /* Version        : v01.2 */
7 /* date           : Mar 25, 2023 */
8 /* ***** */
9 /* ***** HEADER FILES INCLUDES ***** */
10 /* ***** */
11
12 #ifndef STEPPER_PRIV_H_
13 #define STEPPER_PRIV_H_
14
15 /* ***** */
16 /* ***** TYPE_DEF/STRUCT/ENUM SECTION ***** */
17 /* ***** */
18
19 typedef enum{
20     STEPPER_COIL_Reset = STEPPER_COIL_NORMAL,
21     STEPPER_COIL_Set = !STEPPER_COIL_Reset
22 }STEPPER_tenuState;
23
24 #define STEPPER_NUM          4u
25 #define STEPPER_Half_Step    0u
26 #define STEPPER_Full_Step    1u
27
28 /* ***** */
29 /* ***** MACRO/DEFINE SECTION ***** */
30 /* ***** */
31
32 /* ***** */
33 /* ***** CONST SECTION ***** */
34 /* ***** */
35
36 /* ***** */
37 /* ***** VARIABLE SECTION ***** */
38 /* ***** */
39
40 /* ***** */
41 /* ***** FUNCTION SECTION ***** */
42 /* ***** */
43
44 #endif /* STEPPER_PRIV_H_ */
45 /***** E N D (STEPPER_priv.h) *****/
```

H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/LBIT_int.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define [BV](#)(bit) (1u<<(bit))
- #define [SET_BIT](#)(REG, bit) ((REG) |= (1u<<(bit)))
- #define [CLR_BIT](#)(REG, bit) ((REG) &= ~(1u<<(bit)))
- #define [TOG_BIT](#)(REG, bit) ((REG) ^= (1u<<(bit)))
- #define [SET_BYTE](#)(REG, bit) ((REG) |= (0xFFu<<(bit)))
- #define [CLR_BYTE](#)(REG, bit) ((REG) &= ~(0xFFu<<(bit)))
- #define [TOG_BYTE](#)(REG, bit) ((REG) ^= (0xFFu<<(bit)))
- #define [SET_MASK](#)(REG, MASK) ((REG) |= (MASK))
- #define [CLR_MASK](#)(REG, MASK) ((REG) &= ~(MASK))
- #define [TOG_MASK](#)(REG, MASK) ((REG) ^= (MASK))
- #define [GET_MASK](#)(REG, MASK) ((REG) & (MASK))
- #define [SET_REG](#)(REG) ((REG) = ~(0u))
- #define [CLR_REG](#)(REG) ((REG) = (0u))
- #define [TOG_REG](#)(REG) ((REG) ^= ~(0u))
- #define [GET_BIT](#)(REG, bit) (((REG)>>(bit)) & 0x01u)
- #define [GET_NIB](#)(REG, bit) (((REG)>>(bit)) & 0x0Fu)
- #define [GET_BYTE](#)(REG, bit) (((REG)>>(bit)) & 0xFFu)
- #define [ASSIGN_BIT](#)(REG, bit, value) ((REG) = ((REG) & ~(0x01u<<(bit))) | (((value) & 0x01u)<<(bit)))
- #define [ASSIGN_NIB](#)(REG, bit, value) ((REG) = ((REG) & ~(0x0Fu<<(bit))) | (((value) & 0x0Fu)<<(bit)))
- #define [ASSIGN_BYTE](#)(REG, bit, value) ((REG) = ((REG) & ~(0xFFu<<(bit))) | (((value) & 0xFFu)<<(bit)))
- #define [CON_u8Bits](#)(b7, b6, b5, b4, b3, b2, b1, b0)

(0b##b7##b6##b5##b4##b3##b2##b1##b0)
- #define [CON_u16Bits](#)(b15, b14, b13, b12, b11, b10, b9, b8, b7, b6, b5, b4, b3, b2, b1, b0)

(0b##b15##b14##b13##b12##b11##b10##b9##b8##b7##b6##b5##b4##b3##b2##b1##b0)

Macro Definition Documentation

#define _BV(bit) (1u<<(bit))

**#define ASSIGN_BIT(REG, bit, value) ((REG) = ((REG) & ~(0x01u<<(bit))) |
(((value) & 0x01u)<<(bit)))**

**#define ASSIGN_BYTE(REG, bit, value) ((REG) = ((REG) & ~(0xFFu<<(bit))) |
(((value) & 0xFFu)<<(bit)))**

**#define ASSIGN_NIB(REG, bit, value) ((REG) = ((REG) & ~(0x0Fu<<(bit))) |
(((value) & 0x0Fu)<<(bit)))**

#define CLR_BIT(REG, bit) ((REG) &= ~(1u<<(bit)))

#define CLR_BYTE(REG, bit) ((REG) &= ~(0xFFu<<(bit)))

#define CLR_MASK(REG, MASK) ((REG) &= ~(MASK))

#define CLR_REG(REG) ((REG) = (0u))

**#define CON_u16Bits(b15, b14, b13, b12, b11, b10, b9, b8, b7, b6, b5,
b4, b3, b2, b1, b0)**

**(0b##b15##b14##b13##b12##b11##b10##b9##b8##b7##b6##b5##b4##b3##b2##
b1##b0)**

#define CON_u8Bits(b7, b6, b5, b4, b3, b2, b1, b0)

(0b##b7##b6##b5##b4##b3##b2##b1##b0)

#define GET_BIT(REG, bit) (((REG)>>(bit)) & 0x01u)

#define GET_BYTE(REG, bit) (((REG)>>(bit)) & 0xFFu)

#define GET_MASK(REG, MASK) ((REG) & (MASK))

#define GET_NIB(REG, bit) (((REG)>>(bit)) & 0x0Fu)

#define SET_BIT(REG, bit) ((REG) |= (1u<<(bit)))

Bitwise Operation

```
#define SET_BYTE( REG, bit) ((REG) |= (0xFFu<<(bit)))  
  
#define SET_MASK( REG, MASK) ((REG) |= (MASK))  
  
#define SET_REG( REG) ((REG) = ~(0u))  
  
#define TOG_BIT( REG, bit) ((REG) ^= (1u<<(bit)))  
  
#define TOG_BYTE( REG, bit) ((REG) ^= (0xFFu<<(bit)))  
  
#define TOG_MASK( REG, MASK) ((REG) ^= (MASK))  
  
#define TOG_REG( REG) ((REG) ^= ~(0u))
```



```

Go to the documentation of this file.1 /*
***** */
2 /* ***** FILE DEFINITION SECTION ***** */
3 /* ***** */
4 /* File Name      : LBIT_int.h */
5 /* Author         : MAAM */
6 /* Version        : v01 */
7 /* date           : Mar 24, 2023 */
8 /* description    : Bitwise Library */
9 /* ***** */
10 /* ***** HEADER FILES INCLUDES ***** */
11 /* ***** */
12
13 #ifndef LBIT_INT_H_
14 #define LBIT_INT_H_
15
16 /* ***** */
17 /* ***** TYPE_DEF/STRUCT/ENUM SECTION ***** */
18 /* ***** */
19
20 /* ***** */
21 /* ***** MACRO/DEFINE SECTION ***** */
22 /* ***** */
23
24 #define _BV(bit) (1u<<(bit))
25
26
27 #define SET_BIT(REG, bit) ((REG) |= (1u<<(bit)))
28 #define CLR_BIT(REG, bit) ((REG) &= ~(1u<<(bit)))
29 #define TOG_BIT(REG, bit) ((REG) ^= (1u<<(bit)))
30
31 #define SET_BYTE(REG, bit) ((REG) |= (0xFFu<<(bit)))
32 #define CLR_BYTE(REG, bit) ((REG) &= ~(0xFFu<<(bit)))
33 #define TOG_BYTE(REG, bit) ((REG) ^= (0xFFu<<(bit)))
34
35 #define SET_MASK(REG, MASK) ((REG) |= (MASK))
36 #define CLR_MASK(REG, MASK) ((REG) &= ~(MASK))
37 #define TOG_MASK(REG, MASK) ((REG) ^= (MASK))
38 #define GET_MASK(REG, MASK) ((REG) & (MASK))
39
40 #define SET_REG(REG) ((REG) = ~(0u))
41 #define CLR_REG(REG) ((REG) = (0u))
42 #define TOG_REG(REG) ((REG) ^= ~(0u))
43
44 #define GET_BIT(REG, bit) (((REG)>>(bit)) & 0x01u)
45 #define GET_NIB(REG, bit) (((REG)>>(bit)) & 0x0Fu)
46 #define GET_BYTE(REG, bit) (((REG)>>(bit)) & 0xFFu)
47
48 #define ASSIGN_BIT(REG, bit, value) ((REG) = ((REG) & ~(0x01u<<(bit))) | ((value) & 0x01u)<<(bit)))
49 #define ASSIGN_NIB(REG, bit, value) ((REG) = ((REG) & ~(0x0Fu<<(bit))) | ((value) & 0x0Fu)<<(bit)))
50 #define ASSIGN_BYTE(REG, bit, value) ((REG) = ((REG) & ~(0xFFu<<(bit))) | ((value) & 0xFFu)<<(bit)))
51
52 /*
53 #define ASSIGN_BIT(REG,bit,value) do{
54 \
55 \
56 \
57 \
58 \
59 \
60 \
61 \
62 \
63 \
64 \
65 \
66 \
67 \
68 \
69 \
70 \
71 \
72 \
73 \
74 \
75 \
76 \
77 \
78 \
79 \
80 \
81 \
82 \
83 \
84 \
85 \
86 \
87 \
88 \
89 \
90 \
91 \
92 \
93 \
94 \
95 \
96 \
97 \
98 \
99 \
100 \
101 \
102 \
103 \
104 \
105 \
106 \
107 \
108 \
109 \
110 \
111 \
112 \
113 \
114 \
115 \
116 \
117 \
118 \
119 \
120 \
121 \
122 \
123 \
124 \
125 \
126 \
127 \
128 \
129 \
130 \
131 \
132 \
133 \
134 \
135 \
136 \
137 \
138 \
139 \
140 \
141 \
142 \
143 \
144 \
145 \
146 \
147 \
148 \
149 \
150 \
151 \
152 \
153 \
154 \
155 \
156 \
157 \
158 \
159 \
160 \
161 \
162 \
163 \
164 \
165 \
166 \
167 \
168 \
169 \
170 \
171 \
172 \
173 \
174 \
175 \
176 \
177 \
178 \
179 \
180 \
181 \
182 \
183 \
184 \
185 \
186 \
187 \
188 \
189 \
190 \
191 \
192 \
193 \
194 \
195 \
196 \
197 \
198 \
199 \
200 \
201 \
202 \
203 \
204 \
205 \
206 \
207 \
208 \
209 \
210 \
211 \
212 \
213 \
214 \
215 \
216 \
217 \
218 \
219 \
220 \
221 \
222 \
223 \
224 \
225 \
226 \
227 \
228 \
229 \
230 \
231 \
232 \
233 \
234 \
235 \
236 \
237 \
238 \
239 \
240 \
241 \
242 \
243 \
244 \
245 \
246 \
247 \
248 \
249 \
250 \
251 \
252 \
253 \
254 \
255 \
256 \
257 \
258 \
259 \
260 \
261 \
262 \
263 \
264 \
265 \
266 \
267 \
268 \
269 \
270 \
271 \
272 \
273 \
274 \
275 \
276 \
277 \
278 \
279 \
280 \
281 \
282 \
283 \
284 \
285 \
286 \
287 \
288 \
289 \
290 \
291 \
292 \
293 \
294 \
295 \
296 \
297 \
298 \
299 \
300 \
301 \
302 \
303 \
304 \
305 \
306 \
307 \
308 \
309 \
310 \
311 \
312 \
313 \
314 \
315 \
316 \
317 \
318 \
319 \
320 \
321 \
322 \
323 \
324 \
325 \
326 \
327 \
328 \
329 \
330 \
331 \
332 \
333 \
334 \
335 \
336 \
337 \
338 \
339 \
340 \
341 \
342 \
343 \
344 \
345 \
346 \
347 \
348 \
349 \
350 \
351 \
352 \
353 \
354 \
355 \
356 \
357 \
358 \
359 \
360 \
361 \
362 \
363 \
364 \
365 \
366 \
367 \
368 \
369 \
370 \
371 \
372 \
373 \
374 \
375 \
376 \
377 \
378 \
379 \
380 \
381 \
382 \
383 \
384 \
385 \
386 \
387 \
388 \
389 \
390 \
391 \
392 \
393 \
394 \
395 \
396 \
397 \
398 \
399 \
400 \
401 \
402 \
403 \
404 \
405 \
406 \
407 \
408 \
409 \
410 \
411 \
412 \
413 \
414 \
415 \
416 \
417 \
418 \
419 \
420 \
421 \
422 \
423 \
424 \
425 \
426 \
427 \
428 \
429 \
430 \
431 \
432 \
433 \
434 \
435 \
436 \
437 \
438 \
439 \
440 \
441 \
442 \
443 \
444 \
445 \
446 \
447 \
448 \
449 \
450 \
451 \
452 \
453 \
454 \
455 \
456 \
457 \
458 \
459 \
460 \
461 \
462 \
463 \
464 \
465 \
466 \
467 \
468 \
469 \
470 \
471 \
472 \
473 \
474 \
475 \
476 \
477 \
478 \
479 \
480 \
481 \
482 \
483 \
484 \
485 \
486 \
487 \
488 \
489 \
490 \
491 \
492 \
493 \
494 \
495 \
496 \
497 \
498 \
499 \
500 \
501 \
502 \
503 \
504 \
505 \
506 \
507 \
508 \
509 \
510 \
511 \
512 \
513 \
514 \
515 \
516 \
517 \
518 \
519 \
520 \
521 \
522 \
523 \
524 \
525 \
526 \
527 \
528 \
529 \
530 \
531 \
532 \
533 \
534 \
535 \
536 \
537 \
538 \
539 \
540 \
541 \
542 \
543 \
544 \
545 \
546 \
547 \
548 \
549 \
550 \
551 \
552 \
553 \
554 \
555 \
556 \
557 \
558 \
559 \
560 \
561 \
562 \
563 \
564 \
565 \
566 \
567 \
568 \
569 \
570 \
571 \
572 \
573 \
574 \
575 \
576 \
577 \
578 \
579 \
580 \
581 \
582 \
583 \
584 \
585 \
586 \
587 \
588 \
589 \
590 \
591 \
592 \
593 \
594 \
595 \
596 \
597 \
598 \
599 \
600 \
601 \
602 \
603 \
604 \
605 \
606 \
607 \
608 \
609 \
610 \
611 \
612 \
613 \
614 \
615 \
616 \
617 \
618 \
619 \
620 \
621 \
622 \
623 \
624 \
625 \
626 \
627 \
628 \
629 \
630 \
631 \
632 \
633 \
634 \
635 \
636 \
637 \
638 \
639 \
640 \
641 \
642 \
643 \
644 \
645 \
646 \
647 \
648 \
649 \
650 \
651 \
652 \
653 \
654 \
655 \
656 \
657 \
658 \
659 \
660 \
661 \
662 \
663 \
664 \
665 \
666 \
667 \
668 \
669 \
670 \
671 \
672 \
673 \
674 \
675 \
676 \
677 \
678 \
679 \
680 \
681 \
682 \
683 \
684 \
685 \
686 \
687 \
688 \
689 \
690 \
691 \
692 \
693 \
694 \
695 \
696 \
697 \
698 \
699 \
700 \
701 \
702 \
703 \
704 \
705 \
706 \
707 \
708 \
709 \
710 \
711 \
712 \
713 \
714 \
715 \
716 \
717 \
718 \
719 \
720 \
721 \
722 \
723 \
724 \

```

```

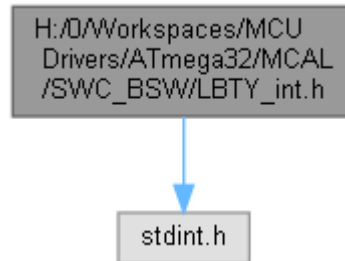
65 (0b##b15##b14##b13##b12##b11##b10##b9##b8##b7##b6##b5##b4##b3##b2##b1##b0)
66
67 /* ***** */
68 /* ***** CONST SECTION ***** */
69 /* ***** */
70
71 /* ***** */
72 /* ***** VARIABLE SECTION ***** */
73 /* ***** */
74
75 /* ***** */
76 /* ***** FUNCTION SECTION ***** */
77 /* ***** */
78
79
80 #endif /* LBIT_INT_H_ */
81 /***** E N D (LBIT_int.h) *****/

```

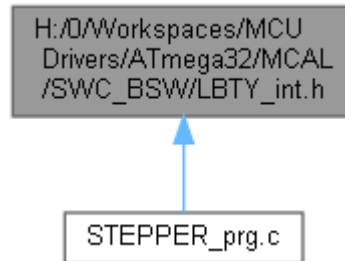
H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/LBTY_int.h File Reference

#include <stdint.h>

Include dependency graph for LBTY_int.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- union [LBTY_tuniPort8](#) union [LBTY_tuniPort16](#)

Macros

- #define [__IO](#) volatile
- #define [__O](#) volatile
- #define [__I](#) volatile const
- #define [LBTY_u8vidNOP](#)()
- #define [LBTY_NULL](#) ((void *) 0U)
- #define [LBTY_u8ZERO](#) ((u8)0x00U)
- #define [LBTY_u8MAX](#) ((u8)0xFFU)
- #define [LBTY_s8MAX](#) ((s8)0x7F)
- #define [LBTY_s8MIN](#) ((s8)0x80)
- #define [LBTY_u16ZERO](#) ((u16)0x0000U)
- #define [LBTY_u16MAX](#) ((u16)0xFFFFU)
- #define [LBTY_s16MAX](#) ((u16)0x7FFF)
- #define [LBTY_s16MIN](#) ((u16)0x8000)
- #define [LBTY_u32ZERO](#) ((u32)0x00000000UL)
- #define [LBTY_u32MAX](#) ((u32)0xFFFFFFFFUL)
- #define [LBTY_s32MAX](#) ((u32)0x7FFFFFFFL)
- #define [LBTY_s32MIN](#) ((u32)0x80000000L)
- #define [LBTY_u64ZERO](#) ((u64)0x0000000000000000ULL)
- #define [LBTY_u64MAX](#) ((u64)0xFFFFFFFFFFFFFFFFULL)
- #define [LBTY_s64MAX](#) ((u64)0x7FFFFFFFFFFFFFFFL)
- #define [LBTY_s64MIN](#) ((u64)0x8000000000000000LL)

Typedefs

- typedef uint8_t [u8](#)
- typedef uint16_t [u16](#)
- typedef uint32_t [u32](#)
- typedef uint64_t [u64](#)
- typedef int8_t [s8](#)
- typedef int16_t [s16](#)
- typedef int32_t [s32](#)
- typedef int64_t [s64](#)
- typedef float [f32](#)
- typedef double [f64](#)
- typedef [u8](#) * [pu8](#)
- typedef [u16](#) * [pu16](#)
- typedef [u32](#) * [pu32](#)
- typedef [u64](#) * [pu64](#)
- typedef [s8](#) * [ps8](#)
- typedef [s16](#) * [ps16](#)
- typedef [s32](#) * [ps32](#)
- typedef [s64](#) * [ps64](#)

Enumerations

- enum [LBTY_tenuFlagStatus](#) { [LBTY_RESET](#) = 0, [LBTY_SET](#) = ![LBTY_RESET](#) }
 - enum [LBTY_tenuBoolean](#) { [LBTY_TRUE](#) = 0x55, [LBTY_FALSE](#) = 0xAA }
 - enum [LBTY_tenuErrorStatus](#) { [LBTY_OK](#) = (u16)0, [LBTY_NOK](#), [LBTY_NULL_POINTER](#), [LBTY_INDEX_OUT_OF_RANGE](#), [LBTY_NO_MASTER_CHANNEL](#), [LBTY_READ_ERROR](#), [LBTY_WRITE_ERROR](#), [LBTY_UNDEFINED_ERROR](#), [LBTY_IN_PROGRESS](#) }
-

Macro Definition Documentation

#define `__I` `volatile const`

#define `__IO` `volatile`

#define `__O` `volatile`

#define `LBTY_NULL` `((void *) 0U)`

#define `LBTY_s16MAX` `((u16)0x7FFF)`

#define `LBTY_s16MIN` `((u16)0x8000)`

#define `LBTY_s32MAX` `((u32)0x7FFFFFFFL)`

#define `LBTY_s32MIN` `((u32)0x80000000L)`

#define `LBTY_s64MAX` `((u64)0x7FFFFFFFFFFFFFFFL)`

#define `LBTY_s64MIN` `((u64)0x8000000000000000LL)`

#define `LBTY_s8MAX` `((s8)0x7F)`

#define `LBTY_s8MIN` `((s8)0x80)`

#define `LBTY_u16MAX` `((u16)0xFFFFU)`

#define `LBTY_u16ZERO` `((u16)0x0000U)`

#define `LBTY_u32MAX` `((u32)0xFFFFFFFFUL)`

#define `LBTY_u32ZERO` `((u32)0x00000000UL)`

#define `LBTY_u64MAX` `((u64)0xFFFFFFFFFFFFFFFFULL)`

#define `LBTY_u64ZERO` `((u64)0x0000000000000000ULL)`

#define `LBTY_u8MAX` `((u8)0xFFU)`

#define `LBTY_u8vidNOP()`

#define `LBTY_u8ZERO` `((u8)0x00U)`

Data Types Limitation

Typedef Documentation

typedef `float` [f32](#)

Standard Real Decimal number

typedef double [f64](#)

typedef [s16](#)* [ps16](#)

typedef [s32](#)* [ps32](#)

typedef [s64](#)* [ps64](#)

typedef [s8](#)* [ps8](#)

Standard Pointer to Signed Byte/Word/Long_Word

typedef [u16](#)* [pu16](#)

typedef [u32](#)* [pu32](#)

typedef [u64](#)* [pu64](#)

typedef [u8](#)* [pu8](#)

Standard Pointer to Unsigned Byte/Word/Long_Word

typedef int16_t [s16](#)

typedef int32_t [s32](#)

typedef int64_t [s64](#)

typedef int8_t [s8](#)

Standard Signed Byte/Word/Long_Word

typedef uint16_t [u16](#)

typedef uint32_t [u32](#)

typedef uint64_t [u64](#)

typedef uint8_t [u8](#)

Data Types New Definitions Standard Unsigned Byte/Word/Long_Word

Enumeration Type Documentation

enum [LBTY_tenuBoolean](#)

Boolean type

Enumerator:

	LBTY_TRUE	
	LBTY_FALSE	

```
96 {
97     LBTY\_TRUE = 0x55,
98     LBTY\_FALSE = 0xAA
99 } LBTY\_tenuBoolean;
```

enum [LBTY_tenuErrorStatus](#)

Error Return type

Enumerator:

LBTY_OK	
LBTY_NOK	
LBTY_NULL_POINTER	
LBTY_INDEX_OUT_OF_RANGE	
LBTY_NO_MASTER_CHANNEL	
LBTY_READ_ERROR	
LBTY_WRITE_ERROR	
LBTY_UNDEFINED_ERROR	
LBTY_IN_PROGRESS	

```
102     {
103     LBTY\_OK = (u16)0,
104     LBTY\_NOK,
105     LBTY\_NULL\_POINTER,
106     LBTY\_INDEX\_OUT\_OF\_RANGE,
107     LBTY\_NO\_MASTER\_CHANNEL,
108     LBTY\_READ\_ERROR,
109     LBTY\_WRITE\_ERROR,
110     LBTY\_UNDEFINED\_ERROR,
111     LBTY\_IN\_PROGRESS          /* Error is not available, wait for availability */
112 } LBTY\_tenuErrorStatus;
```

enum [LBTY_tenuFlagStatus](#)

Flag Status type

Enumerator:

LBTY_RESET	
LBTY_SET	

```
90     {
91     LBTY\_RESET = 0,
92     LBTY\_SET = !LBTY\_RESET
93 } LBTY\_tenuFlagStatus;
```

LBTY_int.h

```
Go to the documentation of this file.1 /*
*****
2 /* ***** FILE DEFINITION SECTION ***** */
3 /* ***** */
4 /* File Name      : LBTY_int.h */
5 /* Author         : MAAM */
6 /* Version        : v01 */
7 /* date           : Mar 23, 2023 */
8 /* description    : Basic Library */
9 /* ***** */
10 /* ***** HEADER FILES INCLUDES ***** */
11 /* ***** */
12
13 #ifndef _LBTY_INT_H_
14 #define _LBTY_INT_H_
15
16 #include <stdint.h>
17
18 /* ***** */
19 /* ***** TYPE_DEF SECTION ***** */
20 /* ***** */
21
22 typedef uint8_t      u8 ;
23 typedef uint16_t     u16;
24 typedef uint32_t     u32;
25 typedef uint64_t     u64;
26
27
28
29 typedef int8_t       s8 ;
30 typedef int16_t      s16;
31 typedef int32_t      s32;
32 typedef int64_t      s64;
33
34
35 typedef float        f32;
36 typedef double       f64;
37
38
39 typedef u8*          pu8 ;
40 typedef u16*         pu16;
41 typedef u32*         pu32;
42 typedef u64*         pu64;
43
44
45 typedef s8*          ps8 ;
46 typedef s16*         ps16;
47 typedef s32*         ps32;
48 typedef s64*         ps64;
49
50
51 /* ***** */
52 /* ***** MACRO/DEFINE SECTION ***** */
53 /* ***** */
54
55 /*****
56 #define __IO      volatile
57 #define __O       volatile
58 #define __I       volatile const
59 *****/
60
61 #define LBTY_u8vidNOP()
62 #define LBTY_NULL      ((void *) 0U)
63
64 #define LBTY_u8ZERO     ((u8)0x00U)
65 #define LBTY_u8MAX      ((u8)0xFFU)
66 #define LBTY_s8MAX      ((s8)0x7F )
67 #define LBTY_s8MIN      ((s8)0x80 )
68
69 #define LBTY_u16ZERO    ((u16)0x0000U)
70 #define LBTY_u16MAX     ((u16)0xFFFFU)
71 #define LBTY_s16MAX     ((u16)0x7FFF )
72 #define LBTY_s16MIN     ((u16)0x8000 )
73
74
75 #define LBTY_u32ZERO    ((u32)0x00000000UL)
76 #define LBTY_u32MAX     ((u32)0xFFFFFFFFUL)
77 #define LBTY_s32MAX     ((u32)0x7FFFFFFF )
78 #define LBTY_s32MIN     ((u32)0x80000000L )
79

```



```

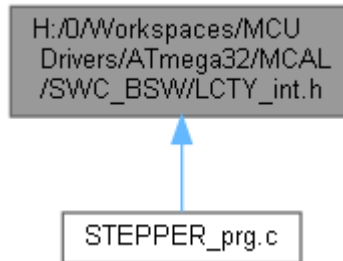
80 #define LBTY_u64ZERO      ((u64)0x0000000000000000ULL)
81 #define LBTY_u64MAX       ((u64)0xFFFFFFFFFFFFFFFFULL)
82 #define LBTY_s64MAX       ((u64)0x7FFFFFFFFFFFFFFFLL )
83 #define LBTY_s64MIN       ((u64)0x8000000000000000LL )
84
85 /* ***** */
86 /* ***** ENUM SECTION ***** */
87 /* ***** */
88
89 typedef enum {
90     LBTY_RESET = 0,
91     LBTY_SET = !LBTY_RESET
92 } LBTY_tenuFlagStatus;
93
94
95 typedef enum {
96     LBTY_TRUE = 0x55,
97     LBTY_FALSE = 0xAA
98 } LBTY_tenuBoolean;
99
100
101 typedef enum {
102     LBTY_OK = (u16)0,
103     LBTY_NOK,
104     LBTY_NULL_POINTER,
105     LBTY_INDEX_OUT_OF_RANGE,
106     LBTY_NO_MASTER_CHANNEL,
107     LBTY_READ_ERROR,
108     LBTY_WRITE_ERROR,
109     LBTY_UNDEFINED_ERROR,
110     LBTY_IN_PROGRESS /* Error is not available, wait for availability */
111 } LBTY_tenuErrorStatus;
112
113
114 /* ***** */
115 /* ***** STRUCT SECTION ***** */
116 /* ***** */
117
118 typedef union {
119     struct {
120         u8 m_u8b0 :1; // LSB
121         u8 m_u8b1 :1;
122         u8 m_u8b2 :1;
123         u8 m_u8b3 :1;
124         u8 m_u8b4 :1;
125         u8 m_u8b5 :1;
126         u8 m_u8b6 :1;
127         u8 m_u8b7 :1; // MSB
128     } sBits;
129     u8 u_u8Byte;
130 } LBTY_tuniPort8;
131
132
133 typedef union {
134     struct {
135         u8 m_u8b0 :1; // LSB
136         u8 m_u8b1 :1;
137         u8 m_u8b2 :1;
138         u8 m_u8b3 :1;
139         u8 m_u8b4 :1;
140         u8 m_u8b5 :1;
141         u8 m_u8b6 :1;
142         u8 m_u8b7 :1;
143         u8 m_u8b8 :1;
144         u8 m_u8b9 :1;
145         u8 m_u8b10 :1;
146         u8 m_u8b11 :1;
147         u8 m_u8b12 :1;
148         u8 m_u8b13 :1;
149         u8 m_u8b14 :1;
150         u8 m_u8b15 :1; // MSB
151     } sBits;
152     struct {
153         u8 m_u8low;
154         u8 m_u8high;
155     } sBytes;
156     u16 u_u16Word;
157 } LBTY_tuniPort16;
158
159 /* ***** */
160 /* ***** FUNCTION SECTION ***** */

```

```
161 /* ***** */
162
163
164 #endif /* _LBTY_INT_H_ */
165 /***** E N D (LBTY_int.h) *****/
```

H:/0/Workspaces/MCU Drivers/ATmega32/MCAL/SWC_BSW/LCTY_int.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define [LCTY_PROGMEM](#) __attribute__((__progmem__))
- #define [LCTY_PURE](#) __attribute__((__pure__))
- #define [LCTY_INLINE](#) __attribute__((always_inline)) static inline
- #define [LCTY_INTERRUPT](#) __attribute__((interrupt))
- #define [CTY_PACKED](#) __attribute__((__packed__))
- #define [LCTY_CONST](#) __attribute__((__const__))
- #define [LCTY_DPAGE](#) __attribute__((dp))
- #define [LCTY_NODPAGE](#) __attribute__((nodp))
- #define [LCTY_SECTION](#)(section) __attribute__((section(# section)))
- #define [LCTY_ASM](#)(cmd) __asm__ __volatile__ (# cmd ::)

Macro Definition Documentation

#define CTY_PACKED __attribute__((__packed__))

#define LCTY_ASM(cmd) __asm__ __volatile__ (# cmd ::)

#define LCTY_CONST __attribute__((__const__))

#define LCTY_DPAGE __attribute__((dp))

#define LCTY_INLINE __attribute__((always_inline)) static inline

#define LCTY_INTERRUPT __attribute__((interrupt))

#define LCTY_NODPAGE __attribute__((nodp))

#define LCTY_PROGMEM __attribute__((__progmem__))

#define LCTY_PURE __attribute__((__pure__))

#define LCTY_SECTION(section) __attribute__((section(# section)))

LCTY_int.h

```
Go to the documentation of this file.1 /*
*****
2 /* ***** FILE DEFINITION SECTION ***** */
3 /* ***** */
4 /* File Name : LCTY_int.h */
5 /* Author : MAAM */
6 /* Version : v00 */
7 /* date : Apr 26, 2023 */
8 /* description : Compiler Library */
9 /* ***** */
10 /* ***** HEADER FILES INCLUDES ***** */
11 /* ***** */
12
13 #ifndef LCTY_INT_H_
14 #define LCTY_INT_H_
15
16 /* ***** */
17 /* ***** TYPE_DEF/STRUCT/ENUM SECTION ***** */
18 /* ***** */
19
20 /* ***** */
21 /* ***** MACRO/DEFINE SECTION ***** */
22 /* ***** */
23
24 /* prog memory attribute */
25 #define LCTY_PROGMEM __attribute__((__progmem__))
26
27 /* pure attribute */
28 #define LCTY_PURE __attribute__((__pure__))
29
30 /* Abstraction for inlining */
31 // #define LCTY_INLINE static inline
32 #define LCTY_INLINE __attribute__((always_inline)) static inline
33
34 /* define function as interrupt handler */
35 #define LCTY_INTERRUPT __attribute__((interrupt))
36
37 /* Memory packed to pass Memory padding */
38 #define CTY_PACKED __attribute__((__packed__))
39
40 /* Const attribute */
41 #define LCTY_CONST __attribute__((__const__))
42
43 /* place variable in direct page */
44 #define LCTY_DPAGE __attribute__((dp))
45
46 /* do not place variable in direct page */
47 #define LCTY_NODPAGE __attribute__((nodp))
48
49 /* Sections */
50 #define LCTY_SECTION(section) __attribute__((section( # section)))
51
52 /* Abstraction for assembly command */
53 #define LCTY_ASM(cmd) __asm__ __volatile__ ( # cmd ::)
54
55 /* ***** */
56 /* ***** CONST SECTION ***** */
57 /* ***** */
58
59 /* ***** */
60 /* ***** VARIABLE SECTION ***** */
61 /* ***** */
62
63 /* ***** */
64 /* ***** FUNCTION SECTION ***** */
65 /* ***** */
66
67
68 #endif /* LCTY_INT_H_ */
69 /***** E N D (LCTY_int.h) *****/
```