

FSM

Generated by Doxygen 1.7.4

Fri Oct 13 2023 15:23:53



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Module Documentation</b>	<b>7</b>
4.1	FSM . . . . .	7
4.1.1	Typedef Documentation . . . . .	8
4.1.1.1	Event . . . . .	8
4.1.1.2	FSM . . . . .	8
4.1.2	Function Documentation . . . . .	8
4.1.2.1	Fsm_Dispatch . . . . .	8
4.1.2.2	Fsm_Init . . . . .	9
4.1.2.3	Fsm_Transition . . . . .	9
4.2	APP . . . . .	9
4.2.1	Function Documentation . . . . .	10
4.2.1.1	main . . . . .	10
4.3	CParser . . . . .	10
4.3.1	Enumeration Type Documentation . . . . .	11
4.3.1.1	HiddenSignals . . . . .	11
4.3.2	Function Documentation . . . . .	11
4.3.2.1	CParser_Init . . . . .	11
<b>5</b>	<b>Class Documentation</b>	<b>13</b>
5.1	CParserFSM Struct Reference . . . . .	13
5.1.1	Detailed Description . . . . .	13
5.2	Event Struct Reference . . . . .	13
5.2.1	Detailed Description . . . . .	14
5.3	FSM Struct Reference . . . . .	14
5.3.1	Detailed Description . . . . .	14
<b>6</b>	<b>File Documentation</b>	<b>15</b>
6.1	CParser.c File Reference . . . . .	15
6.1.1	Detailed Description . . . . .	15

6.2	CParser.h File Reference . . . . .	16
6.2.1	Detailed Description . . . . .	16
6.3	FSM.c File Reference . . . . .	16
6.3.1	Detailed Description . . . . .	17
6.4	FSM.h File Reference . . . . .	17
6.4.1	Detailed Description . . . . .	18
6.5	main.c File Reference . . . . .	18
6.5.1	Detailed Description . . . . .	19

# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

FSM . . . . .	<a href="#">7</a>
APP . . . . .	<a href="#">9</a>
CParser . . . . .	<a href="#">10</a>



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">CParserFSM</a> (Concrete <a href="#">FSM</a> class for CParser ) . . . . .	<a href="#">13</a>
<a href="#">Event</a> (Representation of <a href="#">Event</a> class ) . . . . .	<a href="#">13</a>
<a href="#">FSM</a> (Representation of <a href="#">FSM</a> class ) . . . . .	<a href="#">14</a>





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">CParser.c</a> (This file is the <a href="#">CParserFSM</a> Source file ) . . . . .	15
<a href="#">CParser.h</a> (This file is the <a href="#">CParserFSM</a> header file ) . . . . .	16
<a href="#">FSM.c</a> (This file is the <a href="#">FSM</a> Source file ) . . . . .	16
<a href="#">FSM.h</a> (This file is the <a href="#">FSM</a> Header file ) . . . . .	17
<a href="#">main.c</a> (This file is the <a href="#">FSM</a> Test App Source file ) . . . . .	18



## Chapter 4

# Module Documentation

### 4.1 FSM

#### Classes

- struct [FSM](#)  
*Representation of [FSM](#) class.*
- struct [Event](#)  
*Representation of [Event](#) class.*

#### Files

- file [FSM.h](#)  
*This file is the [FSM](#) Header file.*
- file [FSM.c](#)  
*This file is the [FSM](#) Source file.*

#### Typedefs

- typedef struct [FSM](#) [FSM](#)
- typedef struct [Event](#) [Event](#)
- typedef [FSM\\_STATUS\\_E](#)(\* [StateHandler](#) )([FSM](#) \*me, [Event](#) const \*const evt)

#### Enumerations

- enum [Status\\_Type](#) { [E\\_OK](#), [E\\_NOT\\_OK](#) }  
*Indicate pass/fail of an operation.*
- enum [FSM\\_STATUS\\_E](#) {  
[INIT\\_STATUS](#), [HANDLED\\_STATUS](#), [TRANS\\_STATUS](#), [IGNORED\\_STATUS](#),  
[ERROR\\_STATUS](#) }

Indicate status of *FSM* event processing.

- enum *RESERVED\_SIGNALS\_E* { *INIT\_SIG* = 1, *ENTRY\_SIG*, *EXIT\_SIG*, *USER\_SIG* }

Signals reserved solely for *FSM* use.

## Functions

- *Status\_Type* *Fsm\_Init* (*FSM* \*const *me*, StateHandler initialState)

This function initializes the *FSM* to a known initial state.

- *Status\_Type* *Fsm\_Dispatch* (*FSM* \*const *me*, *Event* const \*const *evt*)

This function is called regularly to dispatch events to the state handlers.

- *FSM\_STATUS\_E* *Fsm\_Transition* (*FSM* \*const *me*, StateHandler nextStateHandler)

This function is called by state handlers to handle transition from one state to another.

### 4.1.1 Typedef Documentation

#### 4.1.1.1 typedef struct *Event* *Event*

Forward declaration of *Event* struct

#### 4.1.1.2 typedef struct *FSM* *FSM*

Forward declaration of *FSM* struct

### 4.1.2 Function Documentation

#### 4.1.2.1 *Status\_Type* *Fsm\_Dispatch* ( *FSM* \*const *me*, *Event* const \*const *evt* )

This function is called regularly to dispatch events to the state handlers.

#### Returns

*Status\_Type* *E\_OK*/*E\_NOT\_OK*

#### Parameters

<i>in</i>	<i>me</i>	instance of <i>FSM</i> to be initialized
<i>in</i>	<i>evt</i>	pointer to event to be passed to the active StateHandler

< Local variables

< Check arguments are not NULL

< Pass event to the statehandler

#### 4.1.2.2 Status\_Type Fsm.Init ( FSM \*const me, StateHandler initialState )

This function initializes the [FSM](#) to a known initial state.

##### Returns

Status\_Type E\_OK/E\_NOT\_OK

##### Parameters

in	me	instance of <a href="#">FSM</a> to be initialized
in	initialState	pointer to StateHandler function for desired <a href="#">FSM</a> initial state

< Local Variables

< Check arguments are not NULL

< Initialize [FSM](#)

< Perform entry action of initial state

#### 4.1.2.3 FSM\_STATUS\_E Fsm.Transition ( FSM \*const me, StateHandler nextStateHandler )

This function is called by state handlers to handle transition from one state to another.

##### Returns

FSM\_STATUS\_E INIT/HANDLED/TRANS/IGNORED/ERROR

##### Parameters

in	me	<a href="#">FSM</a> instance
in	nextState-Handler	Pointer to function for target state

< Local Variables

< Check arguments are not NULL

< Perform exit actions of previous state and entry actions of new state

## 4.2 APP

### Files

- file [main.c](#)

*This file is the [FSM](#) Test App Source file.*

### Functions

- int [main](#) (void)

*Main Application.*

## 4.2.1 Function Documentation

### 4.2.1.1 `int main ( void )`

Main Application.

Includes

#### Returns

`int`

< Local variables

< Get current directory

< Exit if file open failed

< Read file line by line

< Generate events based on file contents

## 4.3 CParser

### Classes

- struct [CParserFSM](#)

*Concrete [FSM](#) class for CParser.*

### Files

- file [CParser.c](#)

*This file is the [CParserFSM](#) Source file.*

- file [CParser.h](#)

*This file is the [CParserFSM](#) header file.*

### Enumerations

- enum [HiddenSignals](#) { **EMPTY\_SIG** }

- enum [CParser\\_State](#) {

**CODE\_STATE, COMMENT\_STATE, SLASH\_STATE, STAR\_STATE,**  
**MAX\_STATES** }

*All possible states of the CParser [FSM](#).*

- enum [CParser\\_Signals](#) { **CHAR\_SIG = USER\_SIG, SLASH\_SIG, STAR\_SIG** }

*CParserFSM-specific events.*

## Functions

- [Status\\_Type CParser\\_Init](#) (CParserFSM \*fsm)  
*CParser Init function.*

### 4.3.1 Enumeration Type Documentation

#### 4.3.1.1 enum HiddenSignals

Includes Reserved Reminder Signal definition

### 4.3.2 Function Documentation

#### 4.3.2.1 [Status\\_Type CParser\\_Init](#) ( CParserFSM \* fsm )

CParser Init function.

#### Returns

Status\_Type E\_OK/E\_NOT\_OK

#### Parameters

<i>in</i>	<i>fsm</i>	<a href="#">FSM</a> instance
-----------	------------	------------------------------

< Initialize [FSM](#)





## Chapter 5

# Class Documentation

### 5.1 CParserFSM Struct Reference

Concrete [FSM](#) class for CParser.

```
#include <CParser.h>
```

#### Public Attributes

- [FSM](#) **super**
- [CParser\\_State](#) **state**

#### 5.1.1 Detailed Description

Concrete [FSM](#) class for CParser.

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

- [CParser.h](#)

### 5.2 Event Struct Reference

Representation of [Event](#) class.

```
#include <FSM.h>
```

#### Public Attributes

- [uint32\\_t](#) **sig**

### 5.2.1 Detailed Description

Representation of [Event](#) class.

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

- [FSM.h](#)

## 5.3 FSM Struct Reference

Representation of [FSM](#) class.

```
#include <FSM.h>
```

### Public Attributes

- StateHandler **stateHandler**

### 5.3.1 Detailed Description

Representation of [FSM](#) class.

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

- [FSM.h](#)

## Chapter 6

# File Documentation

### 6.1 CParser.c File Reference

This file is the [CParserFSM](#) Source file.

```
#include "CParser.h"
#include "stdio.h"
```

#### Enumerations

- enum [HiddenSignals](#) { **EMPTY\_SIG** }

#### Functions

- [Status\\_Type](#) [CParser\\_Init](#) ([CParserFSM](#) \*fsm)  
*CParser Init function.*

#### 6.1.1 Detailed Description

This file is the [CParserFSM](#) Source file.

**Rev:**

**Date:**

**HeadURL:**

end-description

## 6.2 CParser.h File Reference

This file is the [CParserFSM](#) header file.

```
#include "FSM.h"
```

### Classes

- struct [CParserFSM](#)  
*Concrete [FSM](#) class for CParser.*

### Enumerations

- enum [CParser\\_State](#) {  
    **CODE\_STATE**, **COMMENT\_STATE**, **SLASH\_STATE**, **STAR\_STATE**,  
    **MAX\_STATES** }  
*All possible states of the CParser [FSM](#).*
- enum [CParser\\_Signals](#) { **CHAR\_SIG** = **USER\_SIG**, **SLASH\_SIG**, **STAR\_SIG** }  
*CParserFSM-specific events.*

### Functions

- [Status\\_Type](#) [CParser\\_Init](#) ([CParserFSM](#) \*fsm)  
*CParser Init function.*

### 6.2.1 Detailed Description

This file is the [CParserFSM](#) header file.

**Rev:**

**Date:**

**HeadURL:**

end-description

## 6.3 FSM.c File Reference

This file is the [FSM](#) Source file.

```
#include "FSM.h"
```

## Functions

- [Status\\_Type Fsm\\_Init](#) ([FSM](#) \*const me, StateHandler initialState)  
*This function initializes the [FSM](#) to a known initial state.*
- [Status\\_Type Fsm\\_Dispatch](#) ([FSM](#) \*const me, [Event](#) const \*const evt)  
*This function is called regularly to dispatch events to the state handlers.*
- [FSM\\_STATUS\\_E Fsm\\_Transition](#) ([FSM](#) \*const me, StateHandler nextStateHandler)  
*This function is called by state handlers to handle transition from one state to another.*

### 6.3.1 Detailed Description

This file is the [FSM](#) Source file.

**Rev:**

**Date:**

**HeadURL:**

end-description

## 6.4 FSM.h File Reference

This file is the [FSM](#) Header file.

```
#include "stdint.h"
#include "assert.h"
```

## Classes

- struct [FSM](#)  
*Representation of [FSM](#) class.*
- struct [Event](#)  
*Representation of [Event](#) class.*

## Typedefs

- typedef struct [FSM](#) [FSM](#)
- typedef struct [Event](#) [Event](#)
- typedef [FSM\\_STATUS\\_E](#)(\* [StateHandler](#))([FSM](#) \*me, [Event](#) const \*const evt)

## Enumerations

- enum `Status_Type` { `E_OK`, `E_NOT_OK` }  
*Indicate pass/fail of an operation.*
- enum `FSM_STATUS_E` {  
    `INIT_STATUS`, `HANDLED_STATUS`, `TRANS_STATUS`, `IGNORED_STATUS`,  
    `ERROR_STATUS` }  
*Indicate status of `FSM` event processing.*
- enum `RESERVED_SIGNALS_E` { `INIT_SIG` = 1, `ENTRY_SIG`, `EXIT_SIG`, `USER_SIG` }  
*Signals reserved solely for `FSM` use.*

## Functions

- `Status_Type Fsm_Init` (`FSM *const me`, `StateHandler initialState`)  
*This function initializes the `FSM` to a known initial state.*
- `Status_Type Fsm_Dispatch` (`FSM *const me`, `Event const *const evt`)  
*This function is called regularly to dispatch events to the state handlers.*
- `FSM_STATUS_E Fsm_Transition` (`FSM *const me`, `StateHandler nextStateHandler`)  
*This function is called by state handlers to handle transition from one state to another.*

### 6.4.1 Detailed Description

This file is the `FSM` Header file.

**Rev:**

**Date:**

**HeadURL:**

end-description

## 6.5 main.c File Reference

This file is the `FSM` Test App Source file.

```
#include "CParser.h"
#include "stdio.h"
```

```
#include "stdlib.h"
#include <unistd.h>
#include "string.h"
```

## Functions

- int `main` (void)  
*Main Application.*

### 6.5.1 Detailed Description

This file is the [FSM](#) Test App Source file.

**Rev:**

**Date:**

**HeadURL:**

end-description

# Index

- APP, [9](#)
  - main, [10](#)
- CParser, [10](#)
  - CParser\_Init, [11](#)
  - HiddenSignals, [11](#)
- CParser.c, [15](#)
- CParser.h, [16](#)
- CParser\_Init
  - CParser, [11](#)
- CParserFSM, [13](#)
- Event, [13](#)
  - FSM, [8](#)
- FSM, [7](#), [14](#)
  - Event, [8](#)
  - FSM, [8](#)
  - Fsm\_Dispatch, [8](#)
  - Fsm\_Init, [8](#)
  - Fsm\_Transition, [9](#)
- FSM.c, [16](#)
- FSM.h, [17](#)
- Fsm\_Dispatch
  - FSM, [8](#)
- Fsm\_Init
  - FSM, [8](#)
- Fsm\_Transition
  - FSM, [9](#)
- HiddenSignals
  - CParser, [11](#)
- main
  - APP, [10](#)
- main.c, [18](#)