FSM

Generated by Doxygen 1.7.4

Fri Oct 13 2023 15:23:53

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

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

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

in	*me*	instance of FSM to be initialized
in	*evt*	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 FSM to be initialized
in	*initialState*	pointer to StateHandler function for desired FSM initial state

< Local Variables

< Check arguments are not NULL

< Initialize FSM

< Perform entry action of intial 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*	FSM instance
in	*nextStateHandler*	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

`in`	*fsm*	FSM 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