


MPA 1000 sw-master Doxygen documentation

Performed by trainee – Zheleznoy A.S.



□ Main points.

- 1) Analysis of theoretical material
 - 2) Creating a doxyfile configuration file
 - 3) Adding comments directly to the project codes.
- 



Analysis of theoretical material



Doxygen

Creating a doxyfile configuration file

Tag	Appointment	Default
DOXYFILE_ENCODING	The encoding used for all characters in this settings file	UTF-8
OUTPUT_LANGUAGE	Sets the language in which the documentation will be generated	English
PROJECT_NAME	Project name, which can be a single word or a sequence of words (if you are editing outside of Doxywizard, the sequence of words must be enclosed in double quotes)	My Project
PROJECT_NUMBER	This tag can be used to indicate the project number or version	—
PROJECT_BRIEF	A short one-line project description, which is placed at the top of each page and gives an overview of the purpose of the project	—
OUTPUT_DIRECTORY	An absolute or relative path where the documentation will be generated	Текущая директория
INPUT	A space-separated list of files and / or directories that contain the source code of the project	Текущая директория
RECURSIVE	Used when you need to scan source codes in subfolders of specified directories	NO

Adding comments directly to the project codes.


```
1  ////////////////////////////////////////////////////////////////////
2  // Macnica //
3  // 380 Stevens Avenue, Suite 206 //
4  // Solana Beach, CA 92075 //
5  // www.macnica.com //
6  ////////////////////////////////////////////////////////////////////
7  // Host TX Packet Interface (LPQ) API //
8  // //
9  ////////////////////////////////////////////////////////////////////
10 // Version: 0.1 //
11 // Version Date: 27-nov-18 //
12 ////////////////////////////////////////////////////////////////////
13 // Version Date: 27 Nov 2018 //
14 // Author: Marc Levy //
15 // Description : Initial Creation //
16 ////////////////////////////////////////////////////////////////////
17 /**
18 * @file host_pkt_int_tx_api_class.cpp
19 * @author Marc Levy
20 * @brief Initial Creation - Host TX Packet Interface (LPQ) API
21 * @version 0.1
22 * @date 2018-11-27
23 *
24 * @copyright Copyright © 2018 Macnica. All rights reserved.
25 */
```

```
1  ////////////////////////////////////////////////////////////////////
2  // Macnica //
3  // 380 Stevens Avenue, Suite 206 //
4  // Solana Beach, CA 92075 //
5  // www.macnica.com //
6  ////////////////////////////////////////////////////////////////////
7  // Host TX Packet Interface (LPQ) API //
8  // //
9  ////////////////////////////////////////////////////////////////////
10 // Version: 0.1 //
11 // Version Date: 27-nov-18 //
12 ////////////////////////////////////////////////////////////////////
13 // Version Date: 27 Nov 2018 //
14 // Author: Marc Levy //
15 // Description : Initial Creation //
16 ////////////////////////////////////////////////////////////////////
17
18 #include <stdio.h>
19 #include <stdlib.h>
20
21 #include "host_pkt_int_tx_api_class.h"
22
23 /* Net driver access with pread, pwrite */
24 #include <unistd.h>
```

Adding comments directly to the project codes.

```
150
151 /**
152  * @brief Reset function
153  * @return 0 No error, 1 if Hardware I/O Failure, 2 if Hardware check fail, or module not initi
154  */
155 int host_pkt_int_tx_api_class::host_rst(bool rst)
156 {
157     int err;
158     unsigned int ul;
159
160     if (!hw_check_done)
161     {
162         return 2;
163     }
164
165     ul = (unsigned int) rst;
166
167     err = write_reg(HOST_PKT_INT_TX_CTRL, ul);
168
169     if (err)
170     {
171         err = 1;
172     }
173
174     return err;
175 }
176
```

```
141
142 // Reset function
143 // Return Values:
144 // 0: No error
145 // 1: Hardware I/O Failure
146 // 2: Hardware check fail, or module not initialized
147 int host_pkt_int_tx_api_class::host_rst(bool rst)
148 {
149     int err;
150     unsigned int ul;
151
152     if (!hw_check_done)
153     {
154         return 2;
155     }
156
157     ul = (unsigned int) rst;
158
159     err = write_reg(HOST_PKT_INT_TX_CTRL, ul);
160
161     if (err)
162     {
163         err = 1;
164     }
165
166     return err;
167 }
```




Command	Meaning
<code>\authors</code>	Indicates the author or authors
<code>\version</code>	Used to indicate version
<code>\date</code>	Designed to indicate the development date
<code>\bug</code>	List of known bugs
<code>\warning</code>	Warning for use
<code>\copyright</code>	Used license
<code>\example</code>	The command added to the comment to provide a link to the source with the example (added after the command)
<code>\todo</code>	Command used to describe the changes that need to be made (TODO).
<code>\param</code>	<code>\param [<direction>] <name_param> {param_description}</code>
<code>\return</code>	<code>{return_value_description}</code>



Used literature

- 1) <https://www.doxygen.nl/index.html>
- 2) <https://habr.com/ru/post/252101/>
- 3) <https://cutt.ly/hhjlFXE>



Thank you for you
attention!