

```
1
2
3 #ifndef COMMAND_HANDLER_H_
4 #define COMMAND_HANDLER_H_
5
6 #ifndef nullptr
7 #define nullptr ((void *)0)
8 #endif
9
10 #ifndef F_CPU
11 #define F_CPU 16000000UL
12 #endif
13
14 #include <stdbool.h>
15 #include <stdint.h>
16 #include <stdio.h>
17 #include <string.h>
18 #include <stdlib.h>
19 #include <avr/io.h>
20 #include <util/delay.h>
21 #include "nrf24.h"
22
23 #ifndef BIT_MANIPULATION_MACRO
24 #define BIT_MANIPULATION_MACRO 1
25 #define bit_get(p,m) ((p) & (m))
26 #define bit_set(p,m) ((p) |= (m))
27 #define bit_clear(p,m) ((p) &= ~(m))
28 #define bit_flip(p,m) ((p) ^= (m))
29 #define bit_write(c,p,m) (c ? bit_set(p,m) : bit_clear(p,m))
30 #define BIT(x) (0x01 << (x))
31 #define LONGBIT(x) ((unsigned long)0x00000001 << (x))
32 #endif
33
34 #define currentModuleID 0x03
35 #define SOH 0x01
36 #define STX 0x02
37 #define ETX 0x03
38 #define ETB 0x17
39 #define ON_STATE 0xFF
40 #define OFF_STATE 0x00
41
42 typedef struct CommandType {
43     void (*handlerFunction)();
44 } CommandType;
45
46 typedef enum {
47     SUCCESSFUL_DECOMPOSITION,
48     WRONG_HEADER_SEGMENTATION,
49     WRONG_FOOTER_SEGMENTATION,
50     WRONG_CHECKSUM_CONSISTENCY,
51     WRONG_MODULE_ID,
52     UNDEFINED_COMMAND_CODE,
```

```
53     PARAMETER_DATA_OVERFLOW,  
54     PARAMETER_COUNT_OVERSIZE,  
55     RETRANSMISSION_FAILED,  
56     SUCCESSFUL_RETRANSMISSION,  
57     SUCCESSFUL_COMPOSITION  
58 } CommandStatus;  
59  
60  
61 typedef enum {  
62     RF_SUCCESSFUL_TRANSMISSION,  
63     RF_UNREACHABLE_MODULE,  
64     RF_ACKNOWLEDGE_FAILED  
65 } RF_TransmissionStatus;  
66  
67 typedef enum {  
68     UPDATE_ALL_DEVICES_VALUE_ID,  
69     UPDATE_DEVICE_VALUE_ID,  
70     GET_ALL_DEVICES_VALUE_ID,  
71     GET_DEVICE_VALUE_ID,  
72     MESSAGE_STATUS_ID  
73 } CommandTypeID;  
74  
75 typedef struct {  
76     void *startingPointer;  
77     uint8_t byteLength;  
78 } Parameter;  
79  
80 Parameter parameter[12];  
81 uint8_t *command_buffer;  
82 bool memoryInitialized;  
83 uint8_t lastMessagePID;  
84 CommandType lastMessageCommandType;  
85 uint8_t lastTargetModuleID;  
86 uint8_t lastTransmitterModuleID;  
87  
88  
89 #define AVAILABLE_DEVICES 3  
90 uint8_t deviceStoredValue[AVAILABLE_DEVICES]; //Uint8, las posiciones no se ↗  
    guardan en grados  
91  
92  
93  
94 void STRETCHER_POS_CHANGE_HANDLE(uint8_t positionToMove);  
95 void CURTAIN_POS_CHANGE_HANDLE(uint8_t positionToMove);  
96  
97 extern void UPDATE_ALL_DEVICES_VALUE_H(), UPDATE_DEVICE_VALUE_H(), ↗  
    GET_ALL_DEVICES_VALUE_H(), GET_DEVICE_VALUE_H(), MESSAGE_STATUS_H();  
98 extern CommandStatus ComposeMessageToBuffer(CommandTypeID targetTypeID, uint8_t ↗  
    parameterCount, uint8_t targetBoardID);  
99 extern CommandStatus DecomposeMessageFromBuffer();  
100 extern void writeParameterValue(uint8_t parameterIndex, void* parameterData, ↗  
    uint8_t parameterByteLength);
```

```
101 extern void HandleAvailableCommand();
102 extern bool initliazeMemory();
103
104 #endif /* COMMAND_HANDLER_H_ */
```