```
 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
22  #ifndef BIT_MANIPULATION_MACRO
23  #define BIT_MANIPULATION_MACRO 1
24  #define bit_get(p,m) ((p) & (m))
25  #define bit_set(p,m) ((p) |= (m))
26  #define bit_clear(p,m) ((p) &= ~(m))
27  #define bit_flip(p,m) ((p) ^= (m))
28  #define bit_write(c,p,m) (c ? bit_set(p,m) : bit_clear(p,m))
29  #define BIT(x) (0x01 << (x))
30  #define LONGBIT(x) ((unsigned long)0x00000001 << (x))
31  #endif
32
33  typedef struct CommandType {
34      void (*handlerFunction)();
35  } CommandType;
36
37  typedef enum   {
38      SUCCESFUL_DECOMPOSITION,
39      WRONG_HEADER_SEGMENTATION,
40      WRONG_FOOTER_SEGMENTATION,
41      WRONG_CHECKSUM_CONSISTENCY,
42      WRONG_MODULE_ID,
43      UNDEFINED_COMMAND_CODE,
44      PARAMETER_DATA_OVERFLOW,
45      PARAMETER_COUNT_OVERSIZE,
46      RETRANSMISSION_FAILED,
47      SUCCESFUL_RETRANSMISSION,
48      SUCCESFUL_COMPOSITION
49  } CommandStatus;
50
51
52  typedef enum {
```

```
53      RF_SUCCESFUL_TRANSMISSION,
54      RF_UNREACHEABLE_MODULE,
55      RF_ACKNOWLEDGE_FAILED
56  } RF_TransmissionStatus;
57
58  typedef enum   {
59      UPDATE_ALL_DEVICES_VALUE_ID,
60      UPDATE_DEVICE_VALUE_ID,
61      GET_ALL_DEVICES_VALUE_ID,
62      GET_DEVICE_VALUE_ID,
63      MESSAGE_STATUS_ID
64  } CommandTypeID;
65
66  typedef struct {
67      void *startingPointer;
68      uint8_t byteLength;
69  } Parameter;
70
71  typedef enum   {
72      PHONE_MODULE = 0x00,
73      MAIN_MODULE = 0x01,
74      POWER_MODULE = 0x02,
75      MOTOR_MODULE = 0x03,
76  } ModuleInternalCode;
77
78
79  #define currentModuleID MAIN_MODULE
80
81  #define SOH 0x01
82  #define STX 0x02
83  #define ETX 0x03
84  #define ETB 0x17
85  #define ON_STATE    0xFF
86  #define OFF_STATE   0x00
87
88  #define AVAILABLE_DEVICES 4
89  uint16_t device_value[AVAILABLE_DEVICES];
90
91  uint8_t *command_buffer;
92  Parameter parameter[12];
93  bool memoryInitialized;
94
95  uint8_t lastMessagePID;
96  uint8_t lastTargetModuleID;
97  uint8_t lastTransmitterModuleID;
98  CommandType lastMessageCommandType;
99
100 extern bool initliazeMemory();
101 extern void UPDATE_ALL_DEVICES_VALUE_H(), UPDATE_DEVICE_VALUE_H(),
      GET_ALL_DEVICES_VALUE_H(), GET_DEVICE_VALUE_H(), MESSAGE_STATUS_H();
102 extern CommandStatus DecomposeMessageFromBuffer();
103 extern void HandleAvailableCommand();
```

```
104  extern RF_TransmissionStatus RetransmissionToModule();
105  extern CommandStatus ComposeMessageToBuffer(CommandTypeID targetTypeID, uint8_t
       parameterCount, uint8_t targetBoardID);
106  void writeParameterValue(uint8_t parameterIndex, uint8_t* parameterData, uint8_t
       parameterByteLength);
107  void RetransmissionToPhone();
108  #endif /* COMMAND_HANDLER_H_ */
```