```
1
2
3  #ifndef COMMAND_HANDLER_H_
4  #define COMMAND_HANDLER_H_
5
6  #include <stdbool.h>
7  #include <stdint.h>
8
9  #ifndef nullptr
10 #define nullptr ((void *)0)
11 #endif
12
13 #ifndef F_CPU
14 #define F_CPU               16000000UL
15 #endif
16
17 #define AVAILABLE_COMMANDS 6
18 #define COMMAND_BUFFER_SIZE 32
19 #define PARAMETER_BUFFER_SIZE 28
20
21 #ifndef BIT_MANIPULATION_MACRO
22 #define BIT_MANIPULATION_MACRO 1
23 #define bit_get(p,m) ((p) & (m))
24 #define bit_set(p,m) ((p) |= (m))
25 #define bit_clear(p,m) ((p) &= ~(m))
26 #define bit_flip(p,m) ((p) ^= (m))
27 #define bit_write(c,p,m) (c ? bit_set(p,m) : bit_clear(p,m))
28 #define BIT(x) (0x01 << (x))
29 #define LONGBIT(x) ((unsigned long)0x00000001 << (x))
30 #endif
31
32 typedef struct commandType {
33     const char *commandBase;
34     uint8_t nParameters;
35     void (*handlerFunction)();
36 } commandType;
37
38 void *parameter[3];
39 uint8_t *command_buffer;
40 extern bool initliazeMemory();
41 bool memoryInitialized;
42 extern void TURN_RELAY_ON_HANDLE(), TURN_RELAY_OFF_HANDLE(),
     BUILT_IN_LED_TEST_HANDLER(), TURN_EVERYTHING_ON_HANDLER(),
     TURN_EVERYTHING_OFF_HANDLER(), CALL_NURSE_HANDLE();
43
44 extern void composeCommand(void* output_buffer, commandType* commandT, void**
     inputParameter);
45 extern bool decomposeCommand(void* input_buffer, commandType* commandT, void**
     outputParameter);
46
47
48 #endif /* COMMAND_HANDLER_H_ */
```