```
 1
 2 /*
 3  * File:    I2C_ICM42670P_Function.c
 4  * Author:  M.Ricchieri
 5  *
 6  * Created on 10. mai 2023
 7  *
 8  * This code uses the Mc32_I2cUtilCCS.h
 9  */
10
11
12 //-----------------------------------------------------------------------------// Includes
13 #include "stdint.h"
14 #include "Mc32_I2cUtilCCS.h"
15 #include "imu/inv_imu_driver.h"
16 #include "imu/inv_imu_transport.h"
17 #include "I2C_ICM42670P_Functions.h"
18
19
20 //-----------------------------------------------------------------------------// Constants
21 #define ICM42670P_INIT_VALUE 0
22 #define I2C_BUFFER_LEN 10
23 #define ICM42670P_I2C_BUS_WRITE_ARRAY_INDEX 1
24
25
26 // \Brief : Those functions are used by the ICM42670 IMU to communicate with
27 //          the chip in I2C. The first one is used to read and the second to
28 //          write in the chip. Those functions are pointed by the ICM42670
29 //          driver.
30
31 //-----------------------------------------------------------------------------// ICM42670P_I2C_bus_write
32 int ICM42670P_I2C_bus_write(struct inv_imu_serif *serif, uint8_t reg,
33         const uint8_t *buf, uint32_t len){
34
35     int cursor = 0;
36
37     i2c_start();
38     i2c_write(ICM42670P_ADDR_W);
39     i2c_write(reg);
40
41     for (cursor = 0; cursor < len; cursor++){
42
43         i2c_write(*(buf+cursor));
44     }
45     i2c_stop();
46
47     return 0;
48 }
49
50
51 //-----------------------------------------------------------------------------// ICM42670P_I2C_bus_read
52 int ICM42670P_I2C_bus_read(struct inv_imu_serif *serif, uint8_t reg,
53         uint8_t *buf, uint32_t len){
54
55     uint8_t array[len];
56     int cursor = 0;
57
58     i2c_start();
59     i2c_write(ICM42670P_ADDR_W);
60     i2c_write(reg);
61     i2c_start();
62     i2c_write(ICM42670P_ADDR_R);
63
64     for (cursor = 0; cursor < len; cursor++){
65
66         if((cursor+1) < len)
67             // With ACK
68             array[cursor] = i2c_read(1);
69         else
70             // Without ACK
71             array[cursor] = i2c_read(0);
72
73         buf[cursor] = array[cursor];
74         //*(buf+cursor) = array[cursor];
```

```
75      }
76      i2c_stop();
77
78      return 0;
79  }
```