Annexe S.1

```
1   // Copyright (c) Konstantin Belyalov. All rights reserved.
2   // Licensed under the MIT license.
3
4   #ifndef __SHTC3_H
5   #define __SHTC3_H
6
7   #include "main.h"
8
9   #ifdef __cplusplus
10  #define EXPORT extern "C"
11  #else
12  #define EXPORT
13  #endif
14
15
16  // The shtc3 provides a serial number individualized for each device
17  // Params:
18  //  - `hi2c` I2C bus
19  // Returns device id or 0 in case of error.
20  EXPORT uint16_t shtc3_read_id(I2C_HandleTypeDef *hi2c);
21
22  // Put sensor into sleep mode
23  // Params:
24  //  - `hi2c` I2C bus
25  // Returns device id or 0 in case of error.
26  EXPORT uint32_t shtc3_sleep(I2C_HandleTypeDef *hi2c);
27
28  // Wake up sensor.
29  // You must wait for 240us to let sensor enter into IDLE mode.
30  // Params:
31  //  - `hi2c` I2C bus
32  // Returns zero in case of error
33  EXPORT uint32_t shtc3_wakeup(I2C_HandleTypeDef *hi2c);
34
35  // Performs full cycle: starts temperature/humidity measurements using "clock stretch" method.
36  // Params:
37  //  - `hi2c` I2C bus
38  //  - `temp` measured temperature, in C multiplied by 100 (e.g. 24.1C -> 2410)
39  //  - `hum` measured relative humidity, in percents
40  // Returns zero in case of error
41  EXPORT uint32_t shtc3_perform_measurements(I2C_HandleTypeDef *hi2c, int32_t* temp, int32_t* hum);
42
43  // Start temperature/humidity measurements using "clock stretch" approach, in low power mode.
44  // After completed - values can be obtained by shtc3_read_measurements()
45  // Params:
46  //  - `hi2c` I2C bus
47  //  - `temp` measured temperature, in C multiplied by 100 (e.g. 24.1C -> 2410)
48  //  - `hum` measured relative humidity, in percents
49  // Returns zero in case of error
50  EXPORT uint32_t shtc3_perform_measurements_low_power(I2C_HandleTypeDef *hi2c, int32_t* out_temp,
    int32_t* out_hum);
51
52  #endif
53
```