

# HouseKeeping (HK) structure

The HK structure is defined in mission\_hk.c in both the OBC source code and the CSP-term. Please always change both of them together in order to receive the correct HK beacon.

## 1. HK tables collected by the HK beacon

```
#include <util/log.h>
#include <conf_hk.h>
#include <mission_hk.h>
#include <mission_hk_param.h>
#include <speqs.h>
#include <speqs_param.h>
#include <rbeacon.h>
#include <rbeacon_param.h>

#if !ENABLE_HKCOLLECTOR
LOG_GROUP(log_hk, "hk");
#define LOG_DEFAULT log_hk
#endif

int hk_adcs_init_telem1(void) {return 0;}
int hk_adcs_collect_telem1(void) {return 0;}
int hk_adcs_init_telem2(void) {return 0;}
int hk_adcs_collect_telem2(void) {return 0;}

hk_table_t obc_hk_tables[] = {
{
.node = CSP_ADDR_OBC,
.tableid = A3200_BOARD,
.memid = PARAM_INDEX_OBC_BOARD,
.table = a3200_config,
.count = A3200_CONFIG_COUNT,
.size = A3200_MEM_SIZE,
.nodename = "obc",
.param_interval = MISSION_HK_INTV_OBC,
},{
.node = CSP_ADDR_EPS,
.tableid = EPS_HK,
.memid = PARAM_INDEX_EPS,
.table = eps_hk,
.count = EPS_HK_COUNT,
.size = EPS_HK_SIZE,
.nodename = "eps",
.param_interval = MISSION_HK_INTV_EPS,
.collect_fct = hk_eps2_collect,
},{
.node = CSP_ADDR_COM,
.tableid = AX100_PARAM_TELEM,
.memid = PARAM_INDEX_COM,
.table = ax100_telem,
.count = AX100_TELEM_COUNT,
.size = AX100_TELEM_SIZE,
.nodename = "com",
.param_interval = MISSION_HK_INTV_COM,
},{
.node = CSP_ADDR_OBC,
.tableid = PAYLOAD_MEMID,
.memid = PARAM_INDEX_PAYLOAD,
.table = speqs_param,
.count = SPEQS_PARAM_COUNT,
.size = PAYLOAD_PARAM_SIZE,
.nodename = "speqs",
.param_interval = MISSION_HK_INTV_SPEQS,
},{
.node = CSP_ADDR_ADCS,
.tableid = PARAM_ADCS_TELEM1,
.memid = PARAM_ADCS_TELEM1,
.table = adcs_param_telem1,
.count = ADCS_PARAM_TELEM1_COUNT,
.size = ADCS_PARAM_TELEM1_MEMSIZE,
```

```

.nodename = "adcs1",
.param_interval = MISSION_HK_INTV_ADCS_TELEM1,
.init_fct = hk_adcs_init_telem1,
.collect_fct = hk_adcs_collect_telem1,
},{
.node = CSP_ADDR_ADCS,
.tableid = PARAM_ADCS_TELEM2,
.memid = PARAM_ADCS_TELEM2,
.table = adcs_param_telem2,
.count = ADCS_PARAM_TELEM2_COUNT,
.size = ADCS_PARAM_TELEM2_MEMSIZE,
.nodename = "adcs2",
.param_interval = MISSION_HK_INTV_ADCS_TELEM2,
.init_fct = hk_adcs_init_telem2,
.collect_fct = hk_adcs_collect_telem2,
},{
.node = CSP_ADDR_OBC,
.tableid = RBEACON_MEMID,
.memid = PARAM_INDEX_QRNG,
.table = rbeacon_param,
.count = RBEACON_PARAM_COUNT,
.size = RBEACON_PARAM_SIZE,
.nodename = "qrng",
.param_interval = MISSION_HK_INTV_QRNG,
}
};

```

```
/**
```

```
* Power and Temp:
```

```
* This beacon has voltages, currents and temperatures
```

```
*/
```

```
static const uint16_t hk_beacon_format_0[] = {
```

```
/** EPS */
```

```
0x8000 | PARAM_INDEX_EPS,
```

```
EPS_HK_VBOOST(0),
```

```
EPS_HK_VBOOST(1),
```

```
EPS_HK_VBOOST(2),
```

```
EPS_HK_VBATT,
```

```
EPS_HK_CUROUT(0),
```

```
EPS_HK_CUROUT(1),
```

```
EPS_HK_CUROUT(2),
```

```
EPS_HK_CUROUT(3),
```

```
EPS_HK_CUROUT(4),
```

```
EPS_HK_CUROUT(5),
```

```
EPS_HK_CUROUT(6),
```

```
EPS_HK_CURIN(0),
```

```
EPS_HK_CURIN(1),
```

```
EPS_HK_CURIN(2),
```

```
EPS_HK_CURSUN,
```

```
EPS_HK_CURSYS,
```

```
EPS_HK_TEMP(0),
```

```
EPS_HK_TEMP(1),
```

```
EPS_HK_TEMP(2),
```

```
EPS_HK_TEMP(3),
```

```
EPS_HK_TEMP(4),
```

```
EPS_HK_TEMP(5),
```

```
EPS_HK_BATTMODE,
```

```
/** COM */
```

```
0x8000 | PARAM_INDEX_COM,
```

```
AX100_TELEM_TEMP_BRD,
```

```
AX100_TELEM_TEMP_PA,
```

```
AX100_TELEM_LAST_RSSI,
```

```
AX100_TELEM_LAST_RFERR,
```

```
AX100_TELEM_BGND_RSSI,
```

```
/** OBC */
```

```
0x8000 | PARAM_INDEX_OBC_BOARD,
```

```
A3200_CUR_GSSB1,
```

```
A3200_CUR_GSSB2,
```

```
A3200_CUR_FLASH,
```

```
A3200_CUR_PWM,
```

```

A3200_CUR_GPS,
A3200_CUR_WDE,
A3200_TEMP_A,
A3200_TEMP_B,

);

/**
 * Counter beacon:
 * This beacon has counters and bootcauses
 */
static const uint16_t hk_beacon_format_1[] = {

/** EPS */
0x8000 | PARAM_INDEX_EPS,
EPS_HK_WDT_I2C_S,
EPS_HK_WDT_GND_S,
EPS_HK_CNT_BOOT,
EPS_HK_CNT_WDTI2C,
EPS_HK_CNT_WDTGND,
EPS_HK_CNT_WDTCSP(0),
EPS_HK_CNT_WDTCSP(1),
EPS_HK_WDT_CSP_C(0),
EPS_HK_WDT_CSP_C(1),
EPS_HK_BOOTCAUSE,
EPS_HK_LATCHUPS(0),
EPS_HK_LATCHUPS(1),
EPS_HK_LATCHUPS(2),
EPS_HK_LATCHUPS(3),
EPS_HK_LATCHUPS(4),
EPS_HK_LATCHUPS(5),
EPS_HK_OUT_VAL(0),
EPS_HK_OUT_VAL(1),
EPS_HK_OUT_VAL(2),
EPS_HK_OUT_VAL(3),
EPS_HK_OUT_VAL(4),
EPS_HK_OUT_VAL(5),
EPS_HK_OUT_VAL(6),
EPS_HK_OUT_VAL(7),
EPS_HK_PPTMODE,

/** COM */
0x8000 | PARAM_INDEX_COM,
AX100_TELEM_TX_DUTY,
AX100_TELEM_TOT_TX_COUNTER,
AX100_TELEM_TOT_RX_COUNTER,
AX100_TELEM_TOT_TX_BYTES,
AX100_TELEM_TOT_RX_BYTES,
AX100_TELEM_BOOT_COUNTER,
AX100_TELEM_BOOT_CAUSE,
AX100_TELEM_TX_BYTES,
AX100_TELEM_RX_BYTES,
AX100_TELEM_ACTIVE_CONFIG,
AX100_TELEM_TX_COUNTER,
AX100_TELEM_RX_COUNTER,

/** OBC */
0x8000 | PARAM_INDEX_OBC_BOARD,
A3200_PWR_GSSB1,
A3200_PWR_GSSB2,
A3200_PWR_FLASH,
A3200_PWR_PWM,
A3200_PWR_GPS,
A3200_PWR_WDE,
A3200_SWLOAD_COUNT,
A3200_FS_MOUNTED,
A3200_BOOT_COUNTER,
A3200_BOOT_CAUSE,
A3200_CLOCK,

);

/**
 */

```

```

static const uint16_t hk_beacon_format_adcs_0[] = {

0x8000 | PARAM_ADCS_TELEM1,
ADCS_TELEM1_EXTMAG_TEMP,
ADCS_TELEM1_SUNS_TEMP(0),
ADCS_TELEM1_SUNS_TEMP(1),
ADCS_TELEM1_SUNS_TEMP(2),
ADCS_TELEM1_SUNS_TEMP(3),
ADCS_TELEM1_SUNS_TEMP(4),
ADCS_TELEM1_SUNS_TEMP(5),
ADCS_TELEM1_GYRO_TEMP,

ADCS_TELEM1_MAG(0),
ADCS_TELEM1_MAG(1),
ADCS_TELEM1_MAG(2),
ADCS_TELEM1_MAG_VALID,
ADCS_TELEM1_EXTMAG(0),
ADCS_TELEM1_EXTMAG(1),
ADCS_TELEM1_EXTMAG(2),
ADCS_TELEM1_EXTMAG_VALID,
ADCS_TELEM1_SUNS(0),
ADCS_TELEM1_SUNS(1),
ADCS_TELEM1_SUNS(3),
ADCS_TELEM1_SUNS(4),
ADCS_TELEM1_SUNS(5),
ADCS_TELEM1_SUNS_VALID,
ADCS_TELEM1_GYRO(0),
ADCS_TELEM1_GYRO(1),
ADCS_TELEM1_GYRO(2),
ADCS_TELEM1_GYRO_TREND(0),
ADCS_TELEM1_GYRO_TREND(1),
ADCS_TELEM1_GYRO_TREND(2),
ADCS_TELEM1_GYRO_VALID,

ADCS_TELEM1_TORQUERDUTY(0),
ADCS_TELEM1_TORQUERDUTY(1),
ADCS_TELEM1_TORQUERDUTY(2),

0x8000 | PARAM_ADCS_TELEM2,
ADCS_TELEM2_STA_MAG,
ADCS_TELEM2_STA_EXTMAG,
ADCS_TELEM2_STA_CSS,
ADCS_TELEM2_STA_GYRO,
ADCS_TELEM2_STA_BDOT,
ADCS_TELEM2_STA_RUN,
ADCS_TELEM2_TLOOP,
ADCS_TELEM2_MAXTLOOP,

ADCS_TELEM2_BDOT_RATE(0),
ADCS_TELEM2_BDOT_RATE(1),
ADCS_TELEM2_BDOT_DMAG(0),
ADCS_TELEM2_BDOT_DMAG(1),
ADCS_TELEM2_BDOT_DMAG(2),
ADCS_TELEM2_BDOT_DETUMBLED,

ADCS_TELEM2_ACS_MODE,
ADCS_TELEM2_ACS_DMODE,
ADCS_TELEM2_ADS_MODE,
ADCS_TELEM2_ADS_DMODE,
ADCS_TELEM2_EPHEM_MODE,
ADCS_TELEM2_EPHEM_DMODE,
};

static const uint16_t hk_beacon_format_payload_0[] = {
/** SPEQS*/
0x8000 | PARAM_INDEX_PAYLOAD,
PAYLOAD_ENABLED,
SPEQS_EXP_TEMP_UPDATE_RATE,
SPEQS_EXP_TEMP_DELTA_THRESHOLD,
SPEQS_TEMP_THRESHOLD,
SPEQS_FILENAME,
SPEQS_EXP_INDEX,
SPEQS_FORCE_START,

```

```

//New parameters
ENABLED_HEATER,
HEATER_TIMEOUT,
TEMP_SETPT,
DARKCOUNT_TIMEOUT,
FIXED_BASE_LCPR,
LD_CURRENT,
EXP_TIME,
LCPR_REF_INDEX,
LCPR_STEP_SIZE,
LOOP_COUNTER,
APD1_OFFSET,
APD2_OFFSET,
LCPR1_BUF(0),
LCPR1_BUF(1),
LCPR1_BUF(2),
LCPR1_BUF(3),
LCPR2_BUF(0),
LCPR2_BUF(1),
LCPR2_BUF(2),
LCPR2_BUF(3),
};

static const uint16_t hk_beacon_format_payload_1[] = {
/** QRNG*/
0x8000 | PARAM_INDEX_QRNG,
RBEACON_ACTIVE,
RBEACON_UPDATE_PERIOD,
RBEACON_SENDING,
RBEACON_QUEUE_COUNT,
RBEACON_FILEPOSITION,
RBEACON_STARTTIME,
RBEACON_ENDED,
RBEACON_CURRENT_FILENAME,
RBEACON_CURRENT_TIME,
RBEACON_CURRENT_DATA(0),
RBEACON_CURRENT_DATA(1),
RBEACON_CURRENT_DATA(2),
RBEACON_CURRENT_DATA(3),
RBEACON_CURRENT_DATA(4),
RBEACON_CURRENT_DATA(5),
RBEACON_CURRENT_DATA(6),
RBEACON_CURRENT_DATA(7),
RBEACON_CURRENT_DATA(8),
RBEACON_CURRENT_DATA(9),
RBEACON_CURRENT_DATA(10),
RBEACON_CURRENT_DATA(11),
RBEACON_CURRENT_DATA(12),
RBEACON_CURRENT_DATA(13),
RBEACON_CURRENT_DATA(14),
RBEACON_CURRENT_DATA(15),
RBEACON_CURRENT_DATA(16),
RBEACON_CURRENT_DATA(17),
RBEACON_CURRENT_DATA(18),
RBEACON_CURRENT_DATA(19),
RBEACON_CURRENT_DATA(20),
RBEACON_CURRENT_DATA(21),
RBEACON_CURRENT_DATA(22),
RBEACON_CURRENT_DATA(23),
RBEACON_CURRENT_DATA(24),
RBEACON_CURRENT_DATA(25),
RBEACON_CURRENT_DATA(26),
RBEACON_CURRENT_DATA(27),
RBEACON_CURRENT_DATA(28),
RBEACON_CURRENT_DATA(29),
RBEACON_CURRENT_DATA(30),
RBEACON_CURRENT_DATA(31),
RBEACON_PREVIOUS_TIME,
RBEACON_PREVIOUS_DATA(0),
RBEACON_PREVIOUS_DATA(1),
RBEACON_PREVIOUS_DATA(2),
RBEACON_PREVIOUS_DATA(3),

```

```
RBEACON_PREVIOUS_DATA(4),
RBEACON_PREVIOUS_DATA(5),
RBEACON_PREVIOUS_DATA(6),
RBEACON_PREVIOUS_DATA(7),
RBEACON_PREVIOUS_DATA(8),
RBEACON_PREVIOUS_DATA(9),
RBEACON_PREVIOUS_DATA(10),
RBEACON_PREVIOUS_DATA(11),
RBEACON_PREVIOUS_DATA(12),
RBEACON_PREVIOUS_DATA(13),
RBEACON_PREVIOUS_DATA(14),
RBEACON_PREVIOUS_DATA(15),
RBEACON_PREVIOUS_DATA(16),
RBEACON_PREVIOUS_DATA(17),
RBEACON_PREVIOUS_DATA(18),
RBEACON_PREVIOUS_DATA(19),
RBEACON_PREVIOUS_DATA(20),
RBEACON_PREVIOUS_DATA(21),
RBEACON_PREVIOUS_DATA(22),
RBEACON_PREVIOUS_DATA(23),
RBEACON_PREVIOUS_DATA(24),
RBEACON_PREVIOUS_DATA(25),
RBEACON_PREVIOUS_DATA(26),
RBEACON_PREVIOUS_DATA(27),
RBEACON_PREVIOUS_DATA(28),
RBEACON_PREVIOUS_DATA(29),
RBEACON_PREVIOUS_DATA(30),
RBEACON_PREVIOUS_DATA(31),
```

```
};
```

```
hk_beacon_t obc_hk_beacons[] = {
{ .addr = hk_beacon_format_0, .count = sizeof(hk_beacon_format_0) / sizeof(hk_beacon_format_0[0]), .param_interval =
MISSION_HK_BCN_INTERVAL(0), .node = 1, .type = 0 },
{ .addr = hk_beacon_format_1, .count = sizeof(hk_beacon_format_1) / sizeof(hk_beacon_format_1[0]), .param_interval =
MISSION_HK_BCN_INTERVAL(1), .node = 1, .type = 1 },
{ .addr = hk_beacon_format_payload_0, .count = sizeof(hk_beacon_format_payload_0) / sizeof(hk_beacon_format_payload_0[0]),
.param_interval = MISSION_HK_BCN_INTERVAL(2), .node = 1, .type = 2 },
{ .addr = hk_beacon_format_payload_1, .count = sizeof(hk_beacon_format_payload_1) / sizeof(hk_beacon_format_payload_1[0]),
.param_interval = MISSION_HK_BCN_INTERVAL(3), .node = 1, .type = 3 },
{ .addr = hk_beacon_format_adcs_0, .count = sizeof(hk_beacon_format_adcs_0) / sizeof(hk_beacon_format_adcs_0[0]), .param_interval =
MISSION_HK_BCN_INTERVAL(4), .node = 1, .type = 4 },
```

```
};
```