InPlay NanoBeacon™ IN100 Host Controller Mode Application Notes

About Documentation

Document Name	IN100C0 Application Notes	
Part number		
Control Number	IN1IDOC-HC-IN100-EN-V1_0	External User
Revision	V1.0	

Product Status	Document Content	Data Status
In Development	Target specification / MRD	Initial release
Engineering Document	Main functions and features description	Preliminary version
Official Release Document	Description of all functional features	Subject to revision and updates

This document applies to the following products:

Document number	Applicable Products	Document Status
	IN100-D1-R-RC1I	Engineering Document
	IN100-Q1-R-RC1I	Engineering Document
	IN100-D1-R-RC1F	Engineering Document
	IN100-Q1-R-RC1F	Engineering Document

Contents

List of Figures

List of Tables

1. Introduction

The NanoBeacon IN100 device supports a host controller interface, as shown in Figure **1**, and this document will describe how to configure the device with a host controller and the APIs provided for dynamically changing the configuration through this interface.

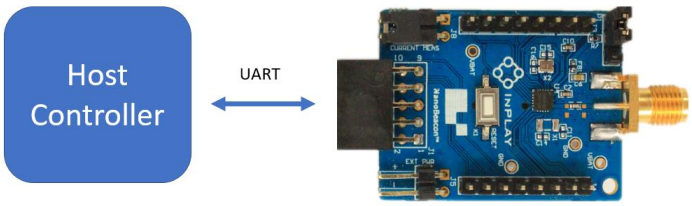

Figure 1: NanoBeacon device in host controller mode

2. Software architecture overview

The software achiecture is as shown in Figure 2, that includes UART communication driver protocol, device register settings and command APIs.

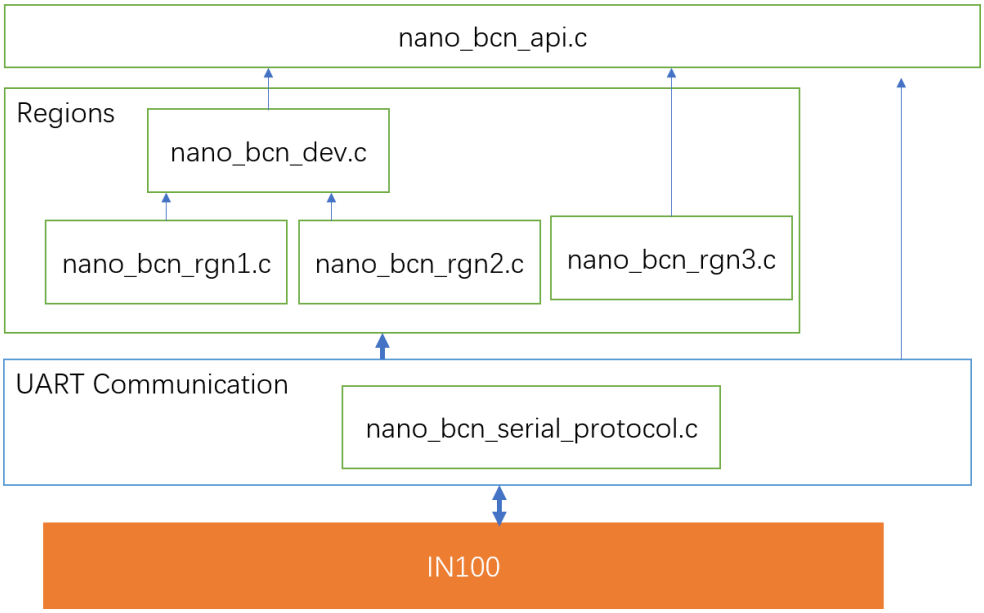

Figure 2 Software architecture

3. Host controller programming API

3.1 UART communication protocol

The external host devices may access the device's internal registers, memory, and eFuse through a UART interface. It supports byte-sequence commands. The general commands and responses format are defined in below table.

Table 1 UART command and data packet format

HEADER (0xAE)	OPCODE (1B)	LENGTH (1B)	PAYLOAD (vary)	TAIL (0xEA)

Table 2 Operation Command Op-Code List

Opcode	Payload Length	Payload Content	Name	Description
0x01	1	0xAA	Ready Request	The first operation command sent to setup connection between master and cicada. The master must receive Ready ACK from IN100 to move on.
0x80	3	Three bytes of rate settings for UART block	Ready ACK	Containing three bytes of rate setting for UART block. Indicating the block is corresponding correctly with the previous command.
0x81	1	One byte error code	Ready NAK	Indicating the block has error processing the previous command. Error code in the payload.
0x02	2	New rate	Rate Change	The rate change command. IN100 should response ready ack with the old rate. But after that all new command should be in new rates.
0x03	2	Address	Reg Read	Register read command
0x83	4	Read data	Reg read resp	Register read value
0x04	6	2-byte address + 4-byte new data	Reg Write	Register write command
0x84	4	4-byte reg data	Reg Write Resp	New register value after register write command
0x05	2	Address	Memory read	Memory read command
0x85	4	Read data	Memory read resp	Memory read value

0x06	6	2-byte address + 4-byte new data	Memory Write	Memory write command
0x86	4	4-byte memory data	Mem write resp	New memory value after memory write command
0x07	1	1-byte word address	eFuse read	eFuse read command
0x87	2	Data	eFuse read resp	eFuse read value
0x08	2	1-byte word address + 1-byte bit address	eFuse write	eFuse write command
0x88	2	Data	eFuse write resp	New eFuse value after the write command at the given address in the payload.

3.2 eFuse region2 definition

The region 2 of the eFuse may contain variable number of words with first word indicating how many words are used by region 2 and the following words contains the real command for the register auto programming. Right after the eFuse Region 1 data.

Commands

4 Types of the command are defined and listed in the following table.

Table 3 Auto programming command

Type value	Description
0	Invalid command. Skip to the next eFuse byte.
1	Delay command
2	Register write command
3	Register read/modify/write command

The first byte of the command always contains the information listed in the following table.

Table 4 First byte of the command

Bit range	Field
1:0	Command type
3:2	Boot condition
7:4	Trigger condition

Time delay command

After the 1st byte, if the command field is 1, the following 3 bytes indicate how many microsecond the command state machine needs to wait before executing the next command. The total length of the time delay command is 4-byte.

Register-write command

After the 1st byte, if the command field is 2, the following several bytes indicate how to write a value into a given register address (based on the size of the value).

Table 5 List of subfields for value field in write command

Bit Index	Description
[1:0]	Size: 0 – byte, 1 – half-word, 2/3 - word
[15:2]	Address to program. Based on the size, the address can be byte address, half-word address, or word address.
[N:16]	Data to be written. If Size is 0: N = 23. If size is 1, N = 31, If size is 2, N = 47.

Based on the size specified in the 2nd byte of the command, the total length of the command can be:

1. 4 bytes if the size is 0 (single byte). The address is byte address and two LSB's in the address field indicate the byte offset in the word.
2. 5 bytes if the size is 1 (half word or 2 bytes). The address is half-word address and the 2nd LSB in the address field indicates the half-word offset in the word. The LSB will be ignored.
3. 7 bytes if the size is 2 or 3 (1 word or 4 bytes). The address is word aligned address. The 2 LSB's will be ignored.

Read-modify-write command

After the 1st byte, if the command field is 3, the following several bytes indicates how to read-modify-write a value into a given register address (based on the size of the value).

Table 6 List of subfields for value Field in Read/Modify/Write command

Bit Index	Description
[1:0]	Size: 0 – byte, 1 – half-word, 2/3 - word
[15:2]	Address to program. Based on the size, the address can be byte address, half-word address, or word address.
[N:16]	Mask for comparison. If Size is 0: N = 23. If size is 1, N = 31, If size is 2, N = 47. If any bit in the mask is 0, the bit will not be modified.
[M:N+1]	Data to be written. If size is 0: M= 31. If size is 1, M = 47. If size is 2 or 3, M = 79.

Based on the size specified in the 2nd byte of the command, the total length of the command can be:

1. 5 bytes if the size is 0 (single byte). The address is byte address and two LSB's in the address field indicate the byte offset in the word.
2. 7 bytes if the size is 1 (half word or 2 bytes). The address is half-word address and the 2nd LSB in the address field indicates the half-word offset in the word. The LSB will be ignored.
3. 11 bytes if the size is 2 or 3 (1 word or 4 bytes). The address is word aligned address. The 2 LSB's will be ignored.

Boot Condition

The command has to satisfy the boot condition to be executed. The condition is specified by the boot condition field (bit 2 to 3) in the first byte of the command as shown in Table 7. The boot condition values are specified as the following table.

Table 7 List of boot conditions

Type value	Description
0	Cold boot only. This trig command will only be executed during cold boot.
1	Warm boot only. This trig command will only be executed during warm boot.
2/3	Cold/warm boot. This command will be executed during cold or warm boot.

Trigger Condition

The command also has to satisfy the trigger condition to be executed. The condition is specified by the trigger condition field (bit 4 to 7) in the first byte of the command as shown in Table 8. The trigger condition values are specified as the following table.

Table 8 List of trigger conditions

Type value	Description
0	Invalid
1	Wakeup
2	Calibration started
3	Calibration done
4	Baseband Tx EN
5	I2C operation done
6	Delayed Tx En done
7	Baseband TX done
8	All Tx done
9	Prepare to sleep
10	ADC operation start
11	ADC operation done
12	Delayed Tx En start
13	Trigger at pulse detection rising edge
14	Trigger at pulse detection done

3.3 registers

gpio_0_pin_mux_pad_ctrl

- Absolute Base Address 0x A08
- Size 32
- Reset Value 0x 00000720
- Access read-write

Fields of Register gpio_0_pin_mux_pad_ctrl

Field Name	Bit Range	Reset Value	Access Type	Description
pin_mux_sel	[2:0]	0x 0	read-write	Pin mux select
inv	3	0x 0	read-write	Invert the output
mux_0_output	4	0x 0	read-write	Output value for MUX 0 option
mux_0_oen	5	0x 1	read-write	output enable value (active low) for MUX 0 option
pad_oe	8	0x 1	read-write	Pad output enable
pad_ie	9	0x 1	read-write	Pad input enable
pad_pu	10	0x 1	read-write	Pad pull-up enable
pad_pd	11	0x 0	read-write	Pad pull-down enable
pad_pc	12	0x 0	read-write	Pad output strength selection
pad_wakeup_input	29	N/A	read-only	Pad input status (since wakeup)
pad_aon_input	30	N/A	read-only	Pad input status (from AON)
pad_input	31	N/A	read-only	Pad input status (real time)

gpio_1_pin_mux_pad_ctrl

- Absolute Base Address 0x A0C
- Size 32
- Reset Value 0x 00000720
- Access read-write

Fields of Register gpio_1_pin_mux_pad_ctrl

Field Name	Bit Range	Reset Value	Access Type	Description
pin_mux_sel	[2:0]	0x 0	read-write	Pin mux select
inv	3	0x 0	read-write	Invert the output
mux_0_output	4	0x 0	read-write	Output value for MUX 0 option
mux_0_oen	5	0x 1	read-write	output enable value (active low) for MUX 0 option
pad_oe	8	0x 1	read-write	Pad output enable
pad_ie	9	0x 1	read-write	Pad input enable
pad_pu	10	0x 1	read-write	Pad pull-up enable
pad_pd	11	0x 0	read-write	Pad pull-down enable
pad_pc	12	0x 0	read-write	Pad output strength selection
pad_wakeup_input	29	N/A	read-only	Pad input status (since wakeup)
pad_aon_input	30	N/A	read-only	Pad input status (from AON)
pad_input	31	N/A	read-only	Pad input status (real time)

gpio_2_pin_mux_pad_ctrl

- Absolute Base Address 0x A10
- Size 32
- Reset Value 0x 00000720

- Access read-write

Fields of Register gpio_2_pin_mux_pad_ctrl

Field Name	Bit Range	Reset Value	Access Type	Description
pin_mux_sel	[2:0]	0x 0	read-write	Pin mux select
inv	3	0x 0	read-write	Invert the output
mux_0_output	4	0x 0	read-write	Output value for MUX 0 option
mux_0_oen	5	0x 1	read-write	output enable value (active low) for MUX 0 option
pad_oe	8	0x 1	read-write	Pad output enable
pad_ie	9	0x 1	read-write	Pad input enable
pad_pu	10	0x 1	read-write	Pad pull-up enable
pad_pd	11	0x 0	read-write	Pad pull-down enable
pad_pc	12	0x 0	read-write	Pad output strength selection
pad_wakeup_input	29	N/A	read-only	Pad input status (since wakeup)
pad_aon_input	30	N/A	read-only	Pad input status (from AON)
pad_input	31	N/A	read-only	Pad input status

gpio_3_pin_mux_pad_ctrl

- Absolute Base Address 0x A14
- Size 32
- Reset Value 0x 00000720
- Access read-write

Fields of Register gpio_3_pin_mux_pad_ctrl

Field Name	Bit Range	Reset Value	Access Type	Description
pin_mux_sel	[2:0]	0x 0	read-write	Pin mux select
inv	3	0x 0	read-write	Invert the output
mux_0_output	4	0x 0	read-write	Output value for MUX 0 option
mux_0_oen	5	0x 1	read-write	output enable value (active low) for MUX 0 option
pad_oe	8	0x 1	read-write	Pad output enable
pad_ie	9	0x 1	read-write	Pad input enable
pad_pu	10	0x 1	read-write	Pad pull-up enable
pad_pd	11	0x 0	read-write	Pad pull-down enable
pad_pc	12	0x 0	read-write	Pad output strength selection
pad_wakeup_input	29	N/A	read-only	Pad input status (since wakeup)
pad_aon_input	30	N/A	read-only	Pad input status (from AON)
pad_input	31	N/A	read-only	Pad input status

gpio_4_pin_mux_pad_ctrl

- Absolute Base Address 0x A18
- Size 32
- Reset Value 0x 00000720
- Access read-write

Fields of Register gpio_4_pin_mux_pad_ctrl

Field Name	Bit Range	Reset Value	Access Type	Description
pin_mux_sel	[2:0]	0x 0	read-write	Pin mux select
inv	3	0x 0	read-write	Invert the output
mux_0_output	4	0x 0	read-write	Output value for MUX 0 option
mux_0_oen	5	0x 1	read-write	output enable value (active low) for MUX 0 option
pad_oe	8	0x 1	read-write	Pad output enable
pad_ie	9	0x 1	read-write	Pad input enable
pad_pu	10	0x 1	read-write	Pad pull-up enable
pad_pd	11	0x 0	read-write	Pad pull-down enable
pad_pc	12	0x 0	read-write	Pad output strength selection
pad_wakeup_input	29	N/A	read-only	Pad input status (since wakeup)
pad_aon_input	30	N/A	read-only	Pad input status (from AON)
pad_input	31	N/A	read-only	Pad input status

gpio_5_pin_mux_pad_ctrl

- Absolute Base Address 0x A1C
- Size 32
- Reset Value 0x 00000720
- Access read-write

Fields of Register gpio_5_pin_mux_pad_ctrl

Field Name	Bit Range	Reset Value	Access Type	Description
pin_mux_sel	[2:0]	0x 0	read-write	Pin mux select
inv	3	0x 0	read-write	Invert the output
mux_0_output	4	0x 0	read-write	Output value for MUX 0 option
mux_0_oen	5	0x 1	read-write	output enable value (active low) for MUX 0 option
pad_oe	8	0x 1	read-write	Pad output enable
pad_ie	9	0x 1	read-write	Pad input enable
pad_pu	10	0x 1	read-write	Pad pull-up enable
pad_pd	11	0x 0	read-write	Pad pull-down enable
pad_pc	12	0x 0	read-write	Pad output strength selection
pad_wakeup_input	29	N/A	read-only	Pad input status (since wakeup)
pad_aon_input	30	N/A	read-only	Pad input status (from AON)
pad_input	31	N/A	read-only	Pad input status

gpio_6_pin_mux_pad_ctrl

- Absolute Base Address 0x A20
- Size 32
- Reset Value 0x 00000720
- Access read-write

Fields of Register gpio_6_pin_mux_pad_ctrl

Field Name	Bit Range	Reset Value	Access Type	Description
pin_mux_sel	[2:0]	0x 0	read-write	Pin mux select
inv	3	0x 0	read-write	Invert the output
mux_0_output	4	0x 0	read-write	Output value for MUX 0 option
mux_0_oen	5	0x 1	read-write	output enable value (active low) for MUX 0 option
pad_oe	8	0x 1	read-write	Pad output enable
pad_ie	9	0x 1	read-write	Pad input enable
pad_pu	10	0x 1	read-write	Pad pull-up enable
pad_pd	11	0x 0	read-write	Pad pull-down enable
pad_pc	12	0x 0	read-write	Pad output strength selection
pad_wakeup_input	29	N/A	read-only	Pad input status (since wakeup)
pad_aon_input	30	N/A	read-only	Pad input status (from AON)
pad_input	31	N/A	read-only	Pad input status

gpio_7_pin_mux_pad_ctrl

- Absolute Base Address 0x A24
- Size 32
- Reset Value 0x 00000720
- Access read-write

Fields of Register gpio_7_pin_mux_pad_ctrl

Field Name	Bit Range	Reset Value	Access Type	Description
pin_mux_sel	[2:0]	0x 0	read-write	Pin mux select
inv	3	0x 0	read-write	Invert the output
mux_0_output	4	0x 0	read-write	Output value for MUX 0 option
mux_0_oen	5	0x 1	read-write	output enable value (active low) for MUX 0 option
pad_oe	8	0x 1	read-write	Pad output enable
pad_ie	9	0x 1	read-write	Pad input enable
pad_pu	10	0x 1	read-write	Pad pull-up enable
pad_pd	11	0x 0	read-write	Pad pull-down enable
pad_pc	12	0x 0	read-write	Pad output strength selection
pad_wakeup_input	29	N/A	read-only	Pad input status (since wakeup)
pad_aon_input	30	N/A	read-only	Pad input status (from AON)

pad_input	31	N/A	read-only	Pad input status

gpio_le_maskb

LE/maskb control for gpio

- Absolute Base Address 0x 3300
- Size 24
- Reset Value 0x 000000
- Access read-write

Fields of Register gpio_le_maskb

Field Name	Bit Range	Reset Value	Access Type	Description
ctl_gpio_le	[7:0]	0x 0	read-write	latch enable
ctl_gpio_maskb	[23:16]	0x 0	read-write	DI mask

gpio_wakeup_mask

GPIO wakeup mask

- Absolute Base Address 0x 32C4
- Size 13
- Reset Value 0x 0000
- Access read-write

Fields of Register gpio_wakeup_mask

Field Name	Bit Range	Reset Value	Access Type	Description
ctl_gpio_wakeup_mask	[3:0]	0x 0	read-write	Wake-up mask for GPIO [1:0] -> GPIO [1:0]
ctl_ms_gpio_wakeup_mask	[7:4]	0x 0	read-write	
ctl_bod_wakeup_mask	12	0x 0	read-write	

gpio_wakeup_polarity

GPIO wakeup polarity

- Absolute Base Address 0x 32C8
- Size 13
- Reset Value 0x 0000
- Access read-write

Fields of Register gpio_wakeup_polarity

Field Name	Bit Range	Reset Value	Access Type	Description
ctl_gpio_wakeup_polarity	[3:0]	0x 0	read-write	[1:0] -> GPIO [1:0]
ctl_ms_gpio_wakeup_polarity	[7:4]	0x 0	read-write	
ctl_bod_wakeup_polarity	12	0x 0	read-write	

schdlr_gpio_trig_polarity

- Absolute Base Address 0x 1310
- Size 24
- Reset Value 0x 000000
- Access read-write

Fields of Register schdlr_gpio_trig_polarity

Field Name	Bit Range	Reset Value	Access Type	Description
ctl_schdlr_gpio_trig_polarity_adv_set_0	[7:0]	0x 0	read-write	GPIO polarity for ADV set 0. 0 - trigger on 1, 1 - trigger on 0
ctl_schdlr_gpio_trig_polarity_adv_set_1	[15:8]	0x 0	read-write	GPIO polarity for ADV set 1. 0 - trigger on 1, 1 - trigger on 0
ctl_schdlr_gpio_trig_polarity_adv_set_2	[23:16]	0x 0	read-write	GPIO polarity for ADV set 2. 0 - trigger on 1, 1 - trigger on 0

efuse_r23_sel

Efuse/SRAM region 2/3/4 selection

- Absolute Base Address 0x 3390
- Size 32
- Reset Value 0x 00000000
- Access read-write

Fields of Register efuse_r23_sel

Field Name	Bit Range	Reset Value	Access Type	Description
ctl_r2_present_sel_aon	0	0x 0	read-write	Region 2: 0 - use efuse, 1 - use memory preconfigured
ctl_r3_present_sel_aon	1	0x 0	read-write	Region 3: 0 - use efuse, 1 - use memory preconfigured
ctl_r4_present_sel_aon	2	0x 0	read-write	Region 4: 0 - use efuse, 1 - use memory preconfigured
ctl_aon_r2_present	4	0x 0	read-write	If set, memory preconfigured for region 2
ctl_aon_r3_present	5	0x 0	read-write	If set, memory preconfigured for region 3

ctl_aon_r4_present	6	0x 0	read-write	If set, memory preconfigured for region 4
ctl_efuse_r2_cp_skip	8	0x 0	read-write	If set, skip copying data from efuse region 2 to memory (so that use old retention data in memory).
ctl_efuse_r3_cp_skip	9	0x 0	read-write	If set, skip copying data from efuse region 3 to memory (so that use old retention data in memory).
ctl_efuse_r4_cp_skip	10	0x 0	read-write	If set, skip copying data from efuse region 4 to memory (so that use old retention data in memory).
ctl_efuse_r2_cp_num_word_skip	[23:16]	0x 0	read-write	Number of words skipped during efuse R2 copy
ctl_efuse_r3_cp_num_word_skip	[31:24]	0x 0	read-write	Number of words skipped during efuse R3 copy

efuse_r1234_sel_cont

Efuse/SRAM region 2/3 selection

- Absolute Base Address 0x 3394
- Size 9
- Reset Value 0x 000
- Access read-write

Fields of Register efuse_r1234_sel_cont

Field Name	Bit Range	Reset Value	Access Type	Description
ctl_efuse_r4_cp_num_word_skip	[7:0]	0x 0	read-write	Number of words skipped during efuse R$ copy
ctl_r1_data_from_ram	8	0x 0	read-write	Data for region 1 directly from RAM instead of eFuse (mainly for RAM only cases)

mainfsm_fsm_manual_ctrl

MAINFSM manual Tx enable control

- Absolute Base Address 0x 10B0
- Size 24
- Reset Value 0x 123F00
- Access read-write

Fields of Register mainfsm_fsm_manual_ctrl

Field Name	Bit Range	Reset Value	Access Type	Description
ctl_manual_tx_en	0	0x 0	read-write	Enble manual TX
ctl_manual_tx_start	1	0x 0	read-write	Start manual TX (regardless of current time)
ctl_fsm_restart	4	0x 0	read-write	Restart the central FSM (as if it just wakes up).
ctl_fsm_sleep_cancel_min_interval	[15:8]	0x 3f	read-write	Minimal wait time in sleep cancel state before next sleep request (in us)
ctl_fsm_restart_dly_cnt_lmt	[23:16]	0x 12	read-write	Delay time (in number of us) for central FSM after restart

2.4 API description

i. Struct

A data struct is defined as below for transmitting and receiving of the UART communication data and command.

```
typedef struct {
  void (*delay)(uint32_t msec);
  int (*serial_tx)(uint8_t *buf, uint16_t buf_len,  uint32_t tmo);
  int (*serial_rx)(uint8_t *buf, uint16_t buf_len,  uint32_t tmo);
  void (*serial_break)(int on);
} host_itf_t;
```

delay : Delay function pointer, delay in milliseconds
serial_tx: UART data sends function Pointers. Unit of timeout(tmo) is milliseconds
serial_rx: UART data receive function Pointers. Unit of timeout(tmo) is milliseconds
serial_break: UART break on/off function Pointers.

ii. Function

A list of functions can be supported through APIs provided and please refer to below for details with sample function code descriptions.

1) Initialization of the device

```
/**
* Usage:    Used to register host interface function and init
* Params:   p_hif    host interface function
```

```
* Return:      0 indicates success, others indicate failure
*/
int nano_bcn_init(host_itf_t *p_hif);
```

2) De-initialization of the device

```
/**
* Usage:       Used to deinit
* Params:
* Return:
*/
void nano_bcn_deinit(void);
```

3) Hardware related parameters settings
- On-chip capacitor code
- Output Tx power

```
/**
* Usage:       Used to set initial advertising parameters
* Params:      capacitor    on-chip crystal oscillator , 1 ~ 15
*              tx_power     transmission power , -5 ~ +5 dBm
* Return:      0 indicates success, others indicate failure
*/
int nano_bcn_board_setup(uint8_t capacitor, int8_t tx_power);
```

4) Advertising set data settings
- Advertising interval
- Advertising data packet

```
/**
* Usage:       Used to set advertising data
* Params:      bdaddr       bluetooth device address , 6 bytes. cannot be all 0 or all 1
*              interval     broadcast interval, millisecond
*              adv_raw_data  advertising raw data, max length is 31 bytes
*              len       length of adv_raw_data
* Return:      0 indicates success, others indicate failure
*/
int nano_bcn_set_advertising(const uint8_t* bdaddr, int interval, const uint8_t *adv_raw_data, int
len);
```

5) Load raw data to SDRAM

```
/**
 * Usage:   Used to load raw data into SDRAM
 * Params:
 * Return: 0 indicates success, others indicate failure
 */
int nano_bcn_load_data_to_ram(void);
```

6) Start advertising command

```
/**
 * Usage:      Used to start advertising
 * Params:
 * Return:     0 indicates success, others indicate failure
 */
int nano_bcn_start_advertising();
```

7) Reset device command

```
/**
 * Usage:      Used to reset chip by software
 * Params:
 * Return:        0 indicates success, others indicate failure
 */
int nano_bcn_chip_reset(void);
```

8) Sleep command

```
/**
 * Usage:      Used to put the chip into sleep mode by UART break on signal
 * Params:
 * Return:
 */
void nano_bcn_sleep_by_uart_break_on();
```

9) Wake-up command

```
/**
 * Usage:      Used to wakeup chip by UART break off signal
 * Params:
 * Return:
 */
void nano_bcn_wakeup_by_uart_break_off();
```

3.4 Software flow

Figure 3 provides an overview of the software flowchart for initiating an advertising event through the host controller interface. The host controller can change the advertising parameters along with the data packet information at any time through such software flow.

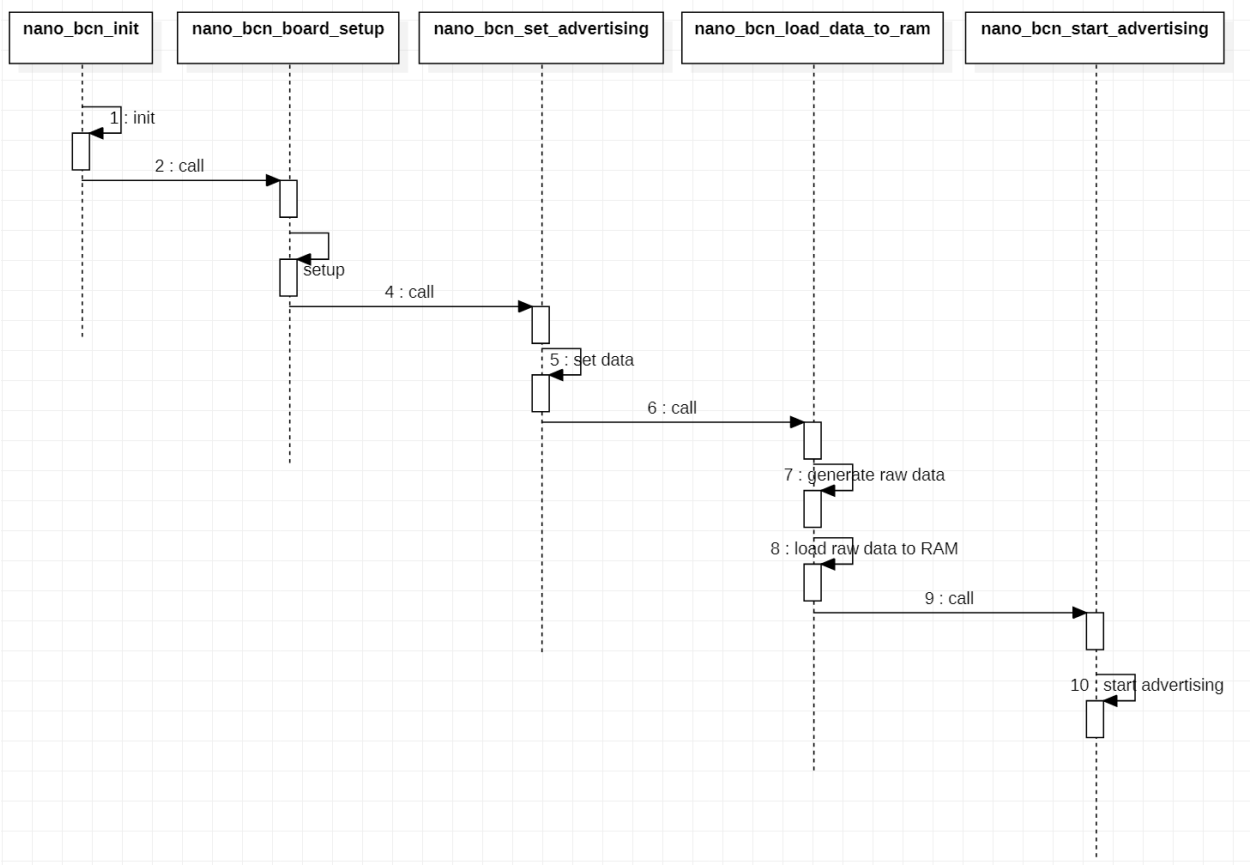

Figure 3 Software flowchart

4. Revision History

Revision	Description	Prepared By	Date
Ver 0.9	Preliminary version	WZ, Liu	2022-05-12
Ver 1.0	Added Register info	WZ, Liu	2022-05-20

5. Disclaimer

InPlay has made every attempt to ensure the accuracy and reliability of the information provided on this document. However, the information is provided "as is" without warranty of any kind. The content of the document will subject to change without prior notice. InPlay does not accept any

responsibility or liability for the accuracy, content, completeness, legally, or reliability of the information contained on this document. We shall not be liable for any loss or damage of whatever nature (direct, indirect, consequential or other) whether arising in contract or otherwise, which may arise as a result of your use of (or inability to use) this document, or from your use of (or failure to use) the information on this document. InPlay Inc and its company logo are registered trademarks of InPlay Inc with its registered office at 1 Technology Drive, STE J728, Irvine, CA, USA 92618.