Basic Communication Manager Design

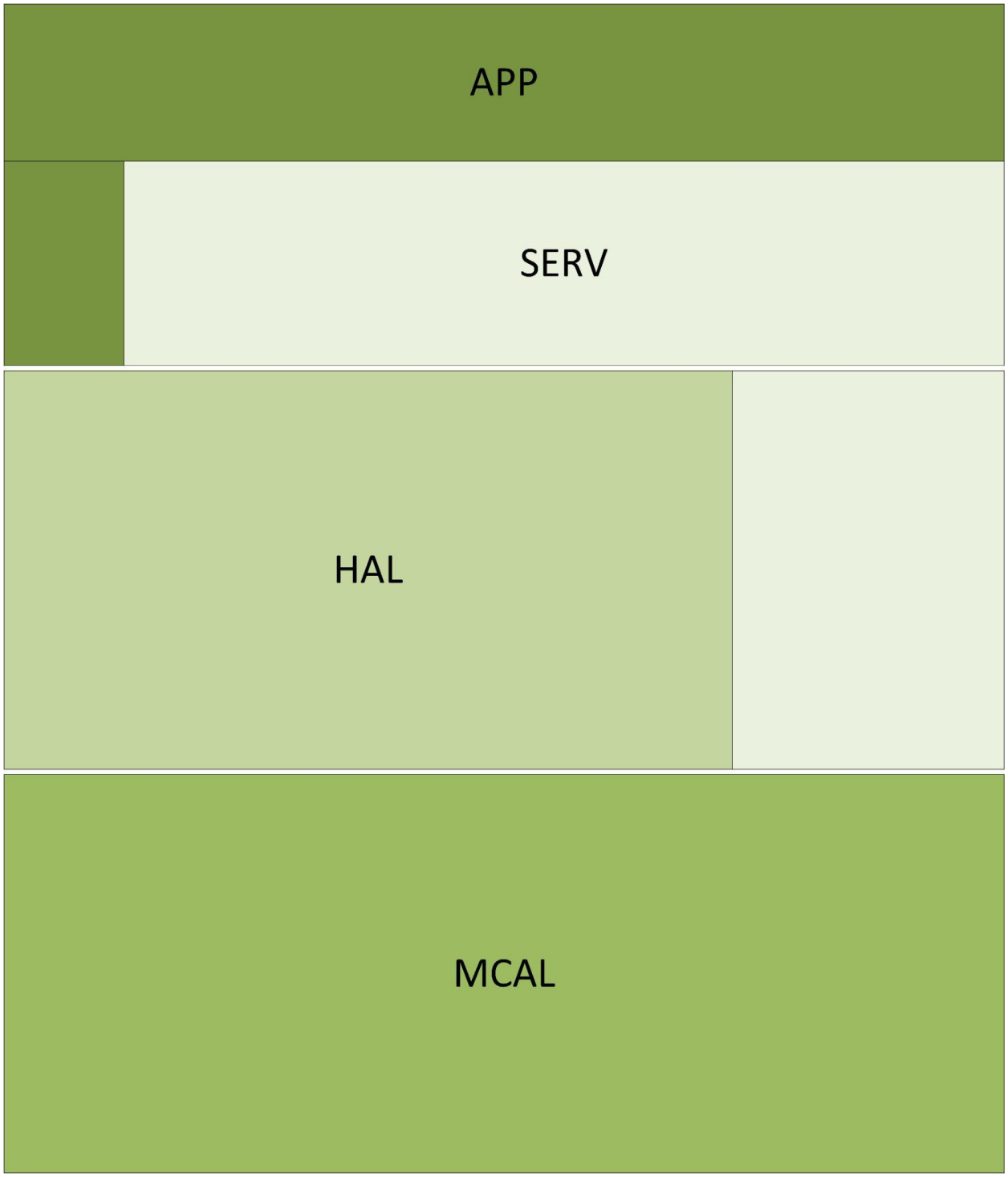
By Mahmoud Sarhan

Table of content:

1 : Detailed Requirements

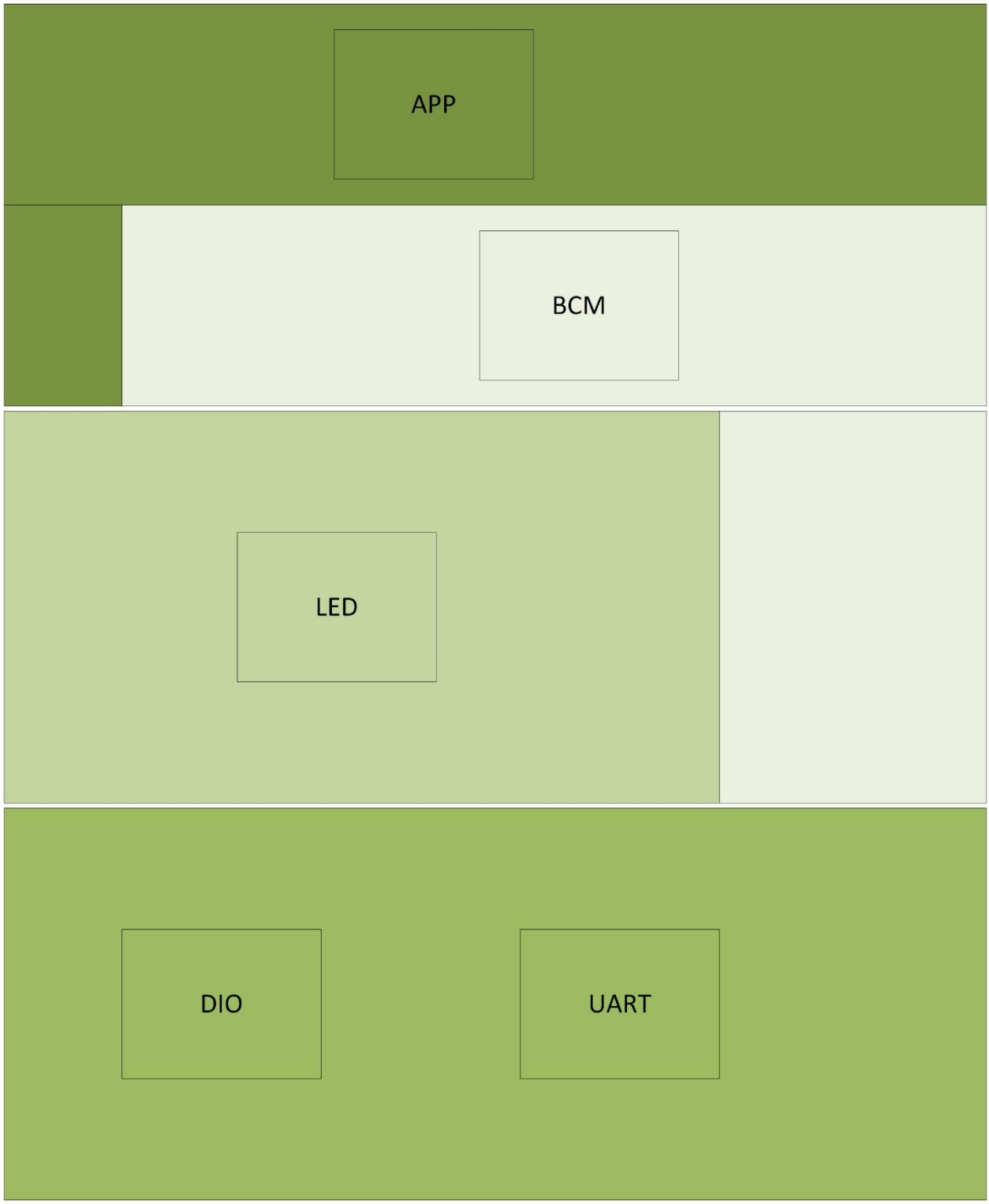
1. Send [BCM Operating] string from MCU_1 to MCU_2.
2. When MCU_1 finish sending, LED_0 in MCU_1 will be toggled.
3. When MCU_2 finish receiving the [BCM Operating] string, LED_1 in MCU_2 will be toggled.
4. MCU_2 will respond with a [Confirm BCM Operating] string to MCU_1.
5. When MCU_2 finish sending, LED_0 in MCU_2 will be toggled.
6. When MCU_1 finish receiving the [BCM Operating] string, LED_1 in MCU_1 will be toggled.

2 : Layered architecture



3 : System modules

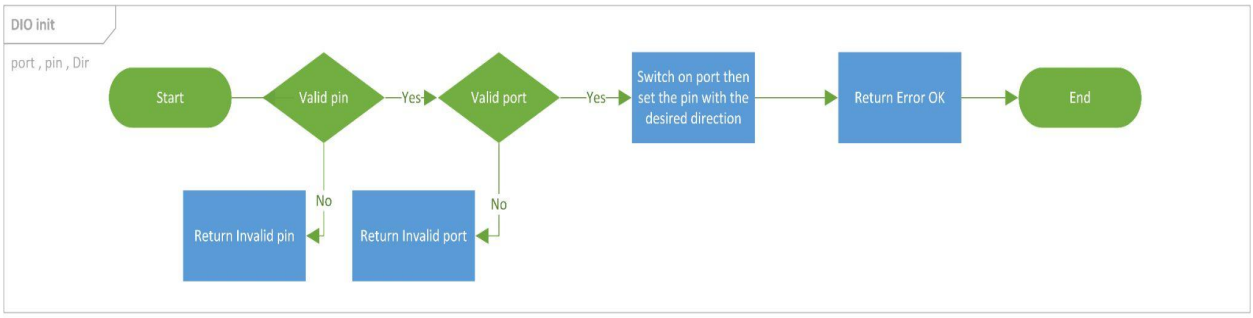
3.1: Module architecture

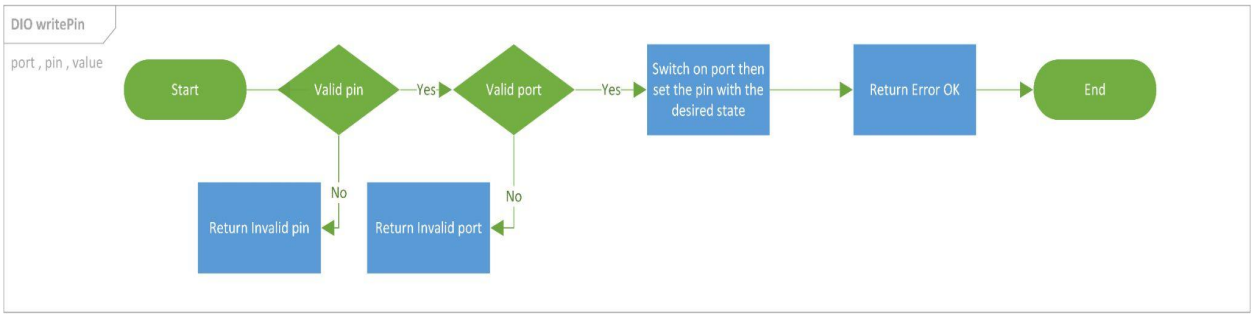


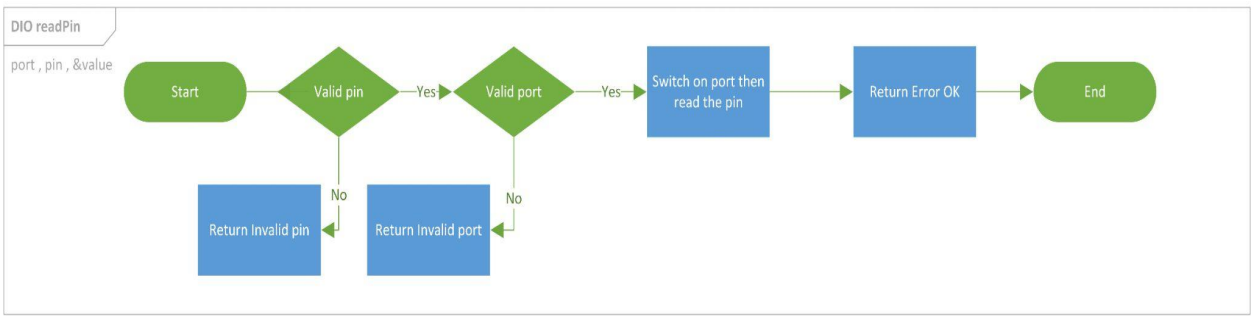
3.2: MCAL APIs

3.2.1: DIO API:

3.2.1.1 :Flowcharts:







3.2.1.2 : Type definitions:

- en_dioPinsType

Name	en_dioPinsType

Type	Enumeration
Range	Shall contain all pins ID
Description	en_dioPinsType
Available via	dio.h

- en_dioPortsType

Name	en_dioPortsType
Type	Enumeration
Range	Shall contain all ports ID
Description	en_dioPortsType
Available via	dio.h

- u8_en_dioErrors

Name	u8_en_dioErrorsType		
Type	Enumeration		
Range	DIO_E_OK	0x00	DIO error OK
	DIO_InvalidPin	0x01	DIO error, invalid pin number.
	DIO_InvalidPort	0x02	DIO error, invalid port number.
Description	u8_en_dioErrors		
Available via	dio.h		

- u8_en_dioLevelType

Name	u8_en_dioLevelType

Type	Enumeration
Range	STD_LOW 0x00 Physical state 0V STD_HIGH 0x01 Physical state 5V or 3.3V.
Description	u8_en_dioLevelType
Available via	dio.h

- u8_en_dioDirType

Name	u8_en_dioDirType
Type	Enumeration
Range	STD_INPUT 0x00 Set pin as input pin STD_OUTPUT 0x01 Set pin as output pin
Description	u8_en_dioDirType
Available via	dio.h

3.2.1.3 : Services affecting the hardware unit:

- DIO_readPIN

Service name	DIO_readPIN		
Syntax	u8_en_dioErrors DIO_readPIN (en_dioPortsType port, en_dioPinsType pin, uint8_t* value);		
Parameters (in)	Port, pin	Channel ID	
	value	Pointer to store the level	STD_HIGH
			STD_LOW

Return	u8_en_dioErrors	DIO_E_OK
		DIO_InvalidPin
		DIO_InvalidPort
Description	This Function gets the level of the pin	

- This function shall return DIO_InvalidPin if pin number is invalid.
- This function shall return DIO_InvalidPort if port number is invalid.

- DIO_writePIN

Service name	DIO_writePIN		
Syntax	u8_en_dioErrors DIO_writePIN (en_dioPortsType port, en_dioPinsType pin, u8_en_dioLevelType state);		
Parameters (in)	Port, pin	Channel ID	
	state	Value to be set	STD_HIGH
			STD_LOW
Return	u8_en_dioErrors	DIO_E_OK	
		DIO_InvalidPin	
		DIO_InvalidPort	
Description	This Function sets the level of the pin		

- This function shall return DIO_InvalidPin if pin number is invalid.
- This function shall return DIO_InvalidPort if port number is invalid.

- DIO_init

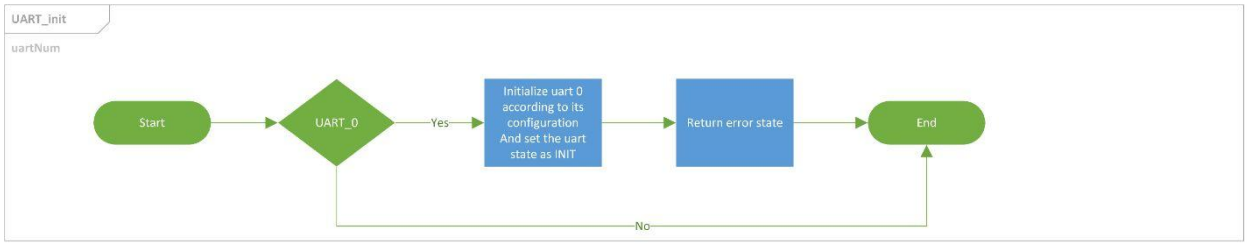
Service name	DIO_init
Syntax	u8_en_dioErrors DIO_init (en_dioPortsType port, en_dioPinsType pin, u8_en_dioDirType direction

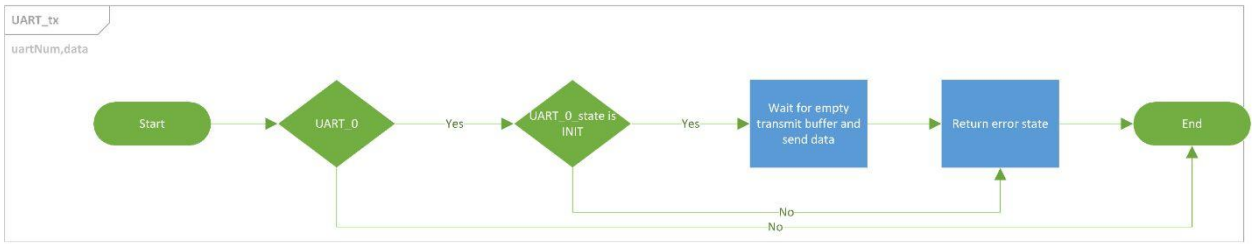
<table>
<tr><td></td><td colspan="3">);</td></tr>
<tr><td rowspan="3">Parameters (in)</td><td>Port, pin</td><td colspan="2">Channel ID</td></tr>
<tr><td rowspan="2">direction</td><td rowspan="2">Value to be set</td><td>STD_INPUT</td></tr>
<tr><td>STD_OUTPUT</td></tr>
<tr><td>Return</td><td colspan="3"><table><tr><td>DIO_Errors</td><td>DIO_E_OK</td></tr><tr><td rowspan="2"></td><td>DIO_InvalidPin</td></tr><tr><td>DIO_InvalidPort</td></tr></table></td></tr>
<tr><td>Description</td><td colspan="3">This Function sets the Direction of the pin</td></tr>
</table>

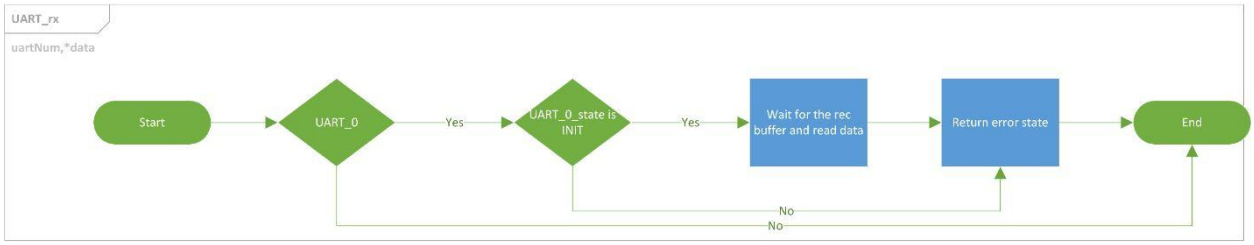
- This function shall return DIO_InvalidPin if pin number is invalid
- This function shall return DIO_InvalidPort if port number is invalid.

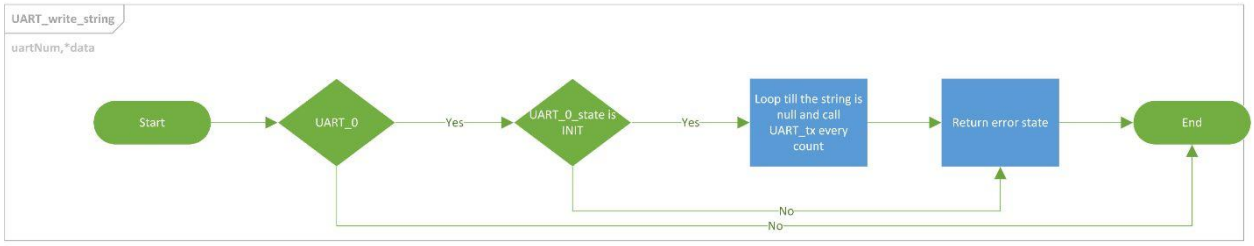
3.2.2: UART API:

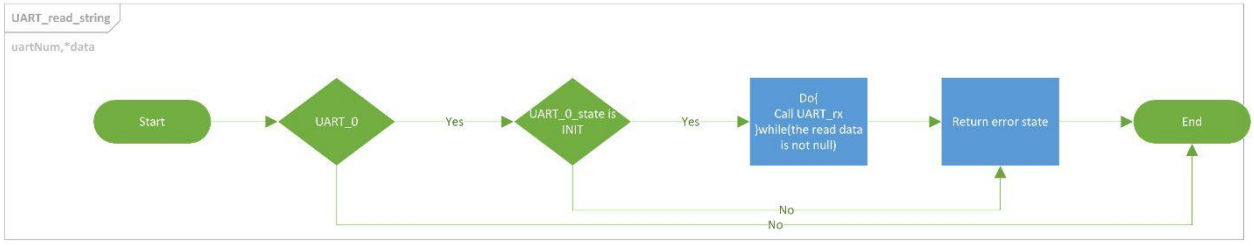
3.2.2.1 :Flowcharts:

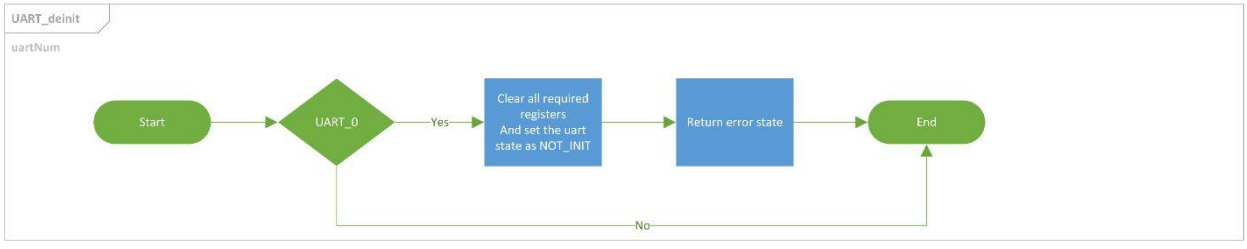

UART_init
uartNum
Start
UART_0
Yes
Initialize uart 0 according to its configuration And set the uart state as INIT
Return error state
End
No


UART_tx
uartNum,data
Start
UART_0
Yes
UART_0_state is INIT
Yes
Wait for empty transmit buffer and send data
Return error state
End
No
No


UART_rx
uartNum,*data
Start
UART_0
Yes
UART_0_state is INIT
Yes
Wait for the rec buffer and read data
Return error state
End
No
No


UART_write_string
uartNum,*data
Start
UART_0
Yes
UART_0_state is INIT
Yes
Loop till the string is null and call UART_tx every count
Return error state
End
No
No


UART_read_string
uartNum,*data
Start
UART_0
Yes
UART_0_state is INIT
Yes
Do{ Call UART_rx }while(the read data is not null)
Return error state
End
No
No


UART_deinit
uartNum
Start
UART_0
Yes
Clear all required registers And set the uart state as NOT_INIT
Return error state
End
No

3.2.1.2 : Type definitions:

- st_uartConfig_t

Name	st_uartConfig_t
Type	Structure
Range	Shall contain required uart configuration
Description	st_uartConfig_t
Available via	uart_pre_link_cfg.h

- u8_uartErrorType

Name	u8_uartErrorType		
Type	Enumeration		
Range	UART_ERROR_OK	0x00	UART error OK
	UART_ERROR_NOT_OK	0x03	UART error
Description	u8_en_timerErrorsType		
Available via	uart.h		

- en_stopBits_t

Name	en_stopBits_t		
Type	Enumeration		
Range	ONE_STOP_BIT	0x00	ONE_STOP_BIT
	TWO_STOP_BIT	0x01	TWO_STOP_BIT
Description	en_stopBits_t		
Available via	uart_pre_link_cfg.h		

- en_dataWidth_t

Name	en_dataWidth_t		
Type	Enumeration		
Range	Data_5_bit	0x00	Data_5_bit
	Data_6_bit	0x01	Data_6_bit
	Data_7_bit	0x02	Data_7_bit
	Data_8_bit	0x03	Data_8_bit
	Data_9_bit	0x07	Data_9_bit
Description	en_dataWidth_t		
Available via	uart_pre_link_cfg.h		

- en_synchronism_t

Name	en_synchronism_t		
Type	Enumeration		
Range	Asynchronous	0x00	Asynchronous
	Synchronous	0x01	Synchronous
Description	en_synchronism_t		
Available via	uart_pre_link_cfg.h		

- en_parity_t

Name	en_parity_t		
Type	Enumeration		
Range	NONE	0x00	NONE
	EVEN	0x02	EVEN
	ODD	0x03	ODD

Description	en_synchronism_t
Available via	uart_pre_link_cfg.h

- en_uartNum_t

Name	en_uartNum_t		
Type	Enumeration		
Range	UART0	0x00	UART0 channel
	INVALID_UART	0x01	Invalid channel
Description	en_uartNum_t		
Available via	uart.h		

3.2.2.3 : Services affecting the hardware unit

- UART_init

Service name	UART_init	
Syntax	u8_uartErrorType UART_init(en_uartNum_t uartNum);	
Parameters (in)	uartNum	UART channel number
Return	u8_uartErrorType	UART_ERROR_OK
		UART_ERROR_NOT_OK
Description	This Function Initialize UART module	

- UART_deinit

Service name	UART_deinit
Syntax	u8_uartErrorType UART_deinit(en_uartNum_t uartNum);

Parameters (in)	uartNum	UART channel number
Return	u8_uartErrorType	UART_ERROR_OK UART_ERROR_NOT_OK
Description	This Function DeInitialize UART module	

- UART_tx

Service name	UART_tx	
Syntax	u8_uartErrorType UART_tx(uint8_t data, en_uartNum_t uartNum);	
Parameters (in)	uartNum	UART channel number
	data	Data to send via uart
Return	u8_uartErrorType	UART_ERROR_OK UART_ERROR_NOT_OK
Description	This Function sends one byte via UART module	

- UART_rx

Service name	UART_rx	
Syntax	u8_uartErrorType UART_rx(uint8_t *data, en_uartNum_t uartNum);	
Parameters (in)	uartNum	UART channel number
	*data	Pointer to save the read data
Return	u8_uartErrorType	UART_ERROR_OK UART_ERROR_NOT_OK
Description	This Function read one byte via UART module	

- UART_write_string

Service name	UART_write_string	
Syntax	u8_uartErrorType UART_write_string(uint8_t *data, en_uartNum_t uartNum);	
Parameters (in)	uartNum	UART channel number
	*data	Pointer to Data to send via uart
Return	u8_uartErrorType	UART_ERROR_OK UART_ERROR_NOT_OK
Description	This Function sends string via UART module	

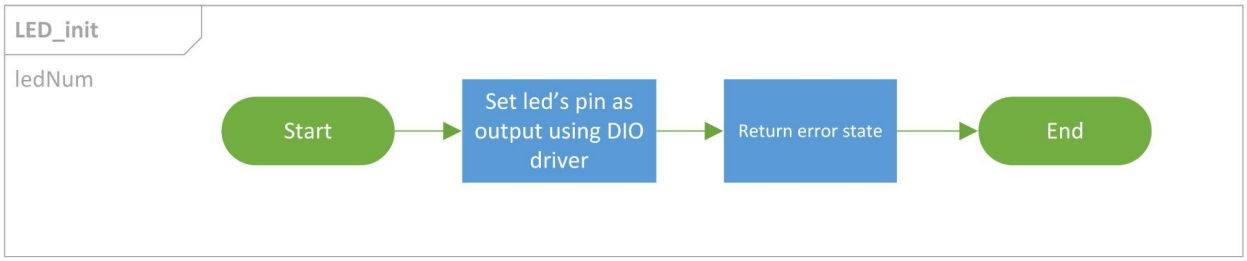
- UART_read_string

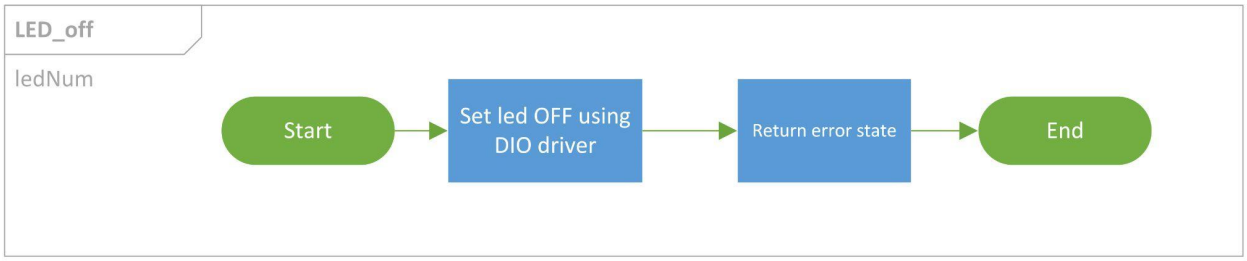
Service name	UART_read_string	
Syntax	u8_uartErrorType UART_read_string(uint8_t *data, en_uartNum_t uartNum);	
Parameters (in)	uartNum	UART channel number
	*data	Pointer to save the read data
Return	u8_uartErrorType	UART_ERROR_OK UART_ERROR_NOT_OK
Description	This Function read string via UART module	

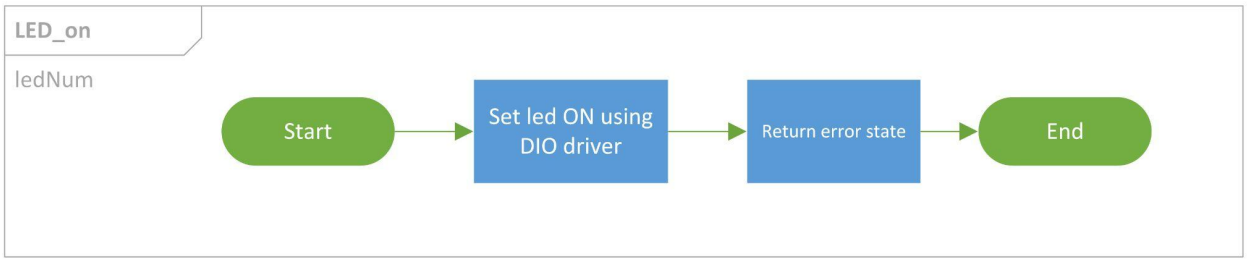
3.3: HAL APIs

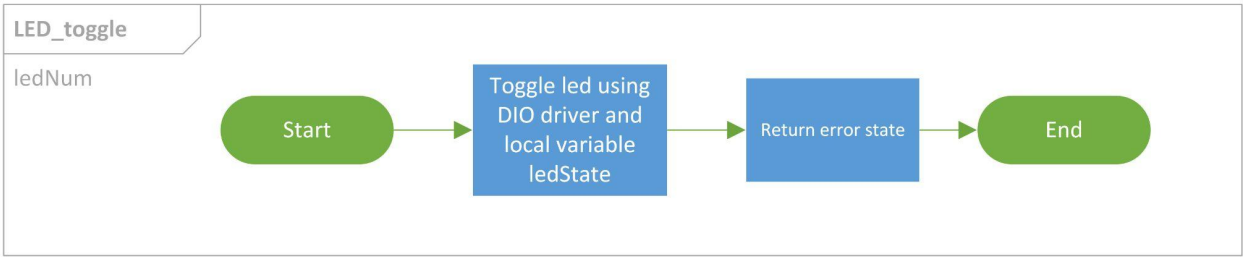
3.3.1: LED API:

3.3.1.1 :Flowcharts:









3.3.1.2 : Type definitions:

- st_ledConfig_t

Name	st_ledConfig_t
Type	Structure
Range	Shall contain required LED configuration
Description	st_ledConfig_t
Available via	led_cfg.h

- u8_ledError_t

Name	u8_ledError_t		
Type	Enumeration		
Range	LED_ERROR_OK	0x00	LED error OK
	LED_ERROR_NOT_OK	0x04	LED error
Description	u8_ledError_t		
Available via	led.h		

- en_ledNum_t

Name	en_ledNum_t		
Type	Enumeration		
Range	LED_0	0x00	LED_0
	LED_1	0x01	LED_1
Description	en_ledNum_t		
Available via	led.h		

3.3.1.3 : Services affecting the hardware unit

- LED_init

Service name	LED_init	
Syntax	u8_ledError_t LED_init(en_ledNum_t ledNum);	
Parameters (in)	ledNum	Led number
Return	u8_ledError_t	LED_ERROR_OK LED_ERROR_NOT_OK
Description	This Function Initialize LED module	

- LED_on

Service name	LED_on	
Syntax	u8_ledError_t LED_on(en_ledNum_t ledNum);	
Parameters (in)	ledNum	Led number
Return	u8_ledError_t	LED_ERROR_OK LED_ERROR_NOT_OK
Description	This Function turn on LED module	

- LED_off

Service name	LED_off	
Syntax	u8_ledError_t LED_off(en_ledNum_t ledNum);	
Parameters (in)	ledNum	Led number
Return	u8_ledError_t	LED_ERROR_OK LED_ERROR_NOT_OK

Description	This Function turn of LED module

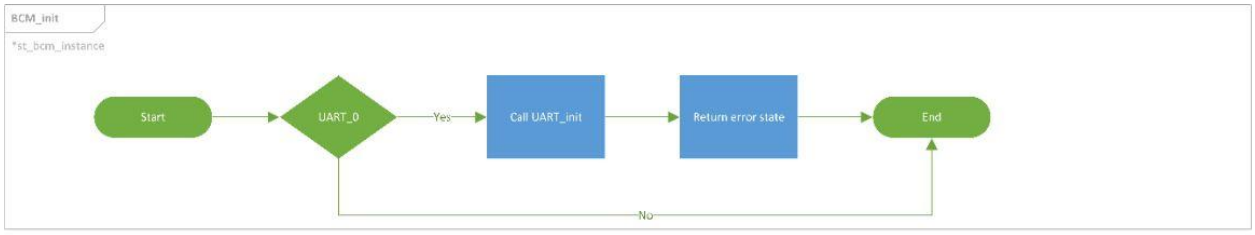
- LED_toggle

Service name	LED_toggle	
Syntax	u8_ledError_t LED_toggle(en_ledNum_t ledNum);	
Parameters (in)	ledNum	Led number
Return	u8_ledError_t	LED_ERROR_OK LED_ERROR_NOT_OK
Description	This Function toggles LED module	

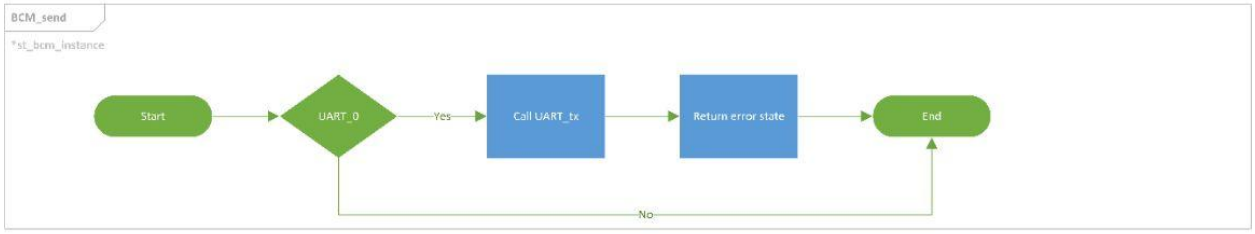
3.4: SERV APIs

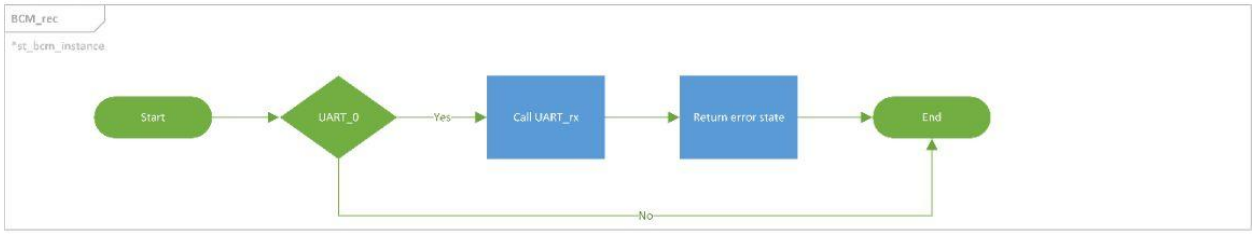
3.4.1: BCM API:

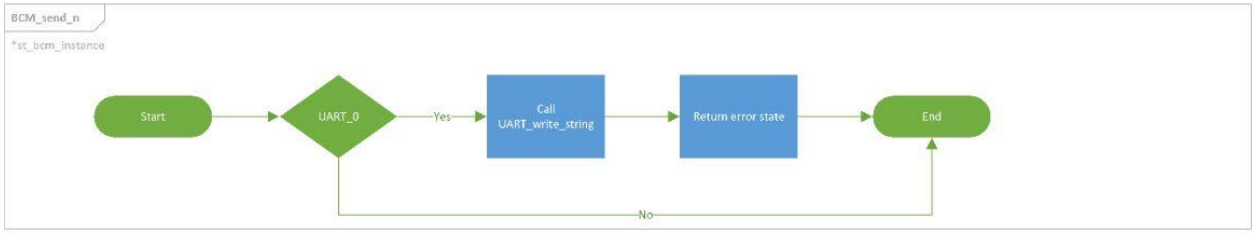
3.4.1.1 :Flowcharts:

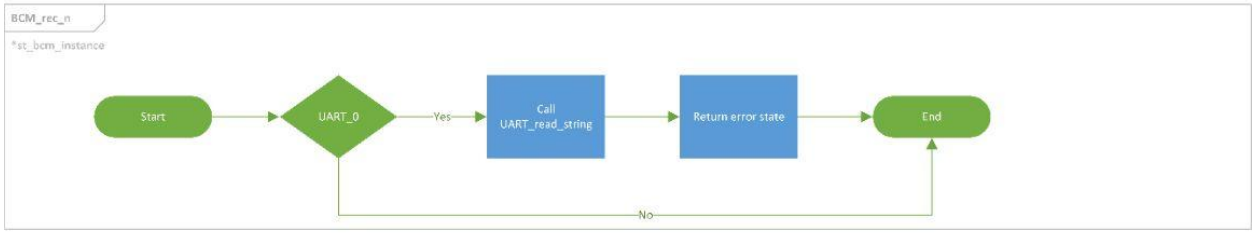

BCM_init
*st_bcm_instance
Start
UART_0
Yes
Call UART_init
Return error state
End
No

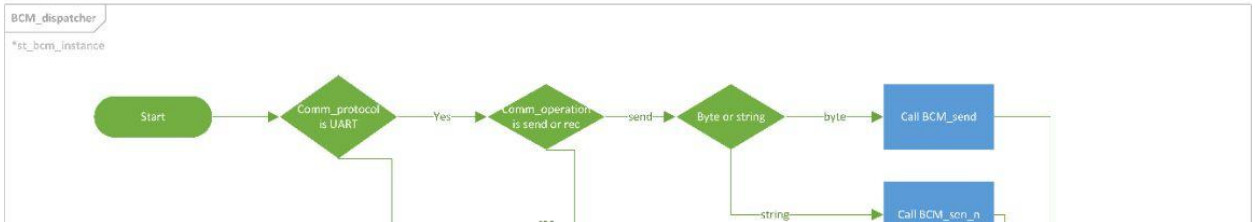

BCM_deinit
*st_bcm_instance
Start
UART_0
Yes
Call UART_deinit
Return error state
End
No


BCM_send
*st_bcm_instance
Start
UART_0
Yes
Call UART_tx
Return error state
End
No


BCM_rec
*st_bcm_instance
Start
UART_0
Yes
Call UART_rx
Return error state
End
No


BCM_send_n
*st_bcm_instance
Start
UART_0
Yes
Call
UART_write_string
Return error state
End
No


BCM_rec_n
*st_bcm_instance
Start
UART_0
Yes
Call
UART_read_string
Return error state
End
No


BCM_dispatcher
*st_bcm_instance
Start
Comm_protocol
is UART
Yes
Comm_operation
is send or rec
send
Byte or string
byte
Call BCM_send
string
Call BCM_scn_n

3.4.1.2 : Type definitions:

- st_bcm_instance_t

Name	st_bcm_instance_t
Type	Structure
Range	Shall contain required BCM configuration
Description	st_bcm_instance_t
Available via	bcm.h

- u8_bcmError_t

Name	u8_bcmError_t		
Type	Enumeration		
Range	BCM_ERROR_OK	0x00	BCM error OK
	BCM_ERROR_NOT_OK	0x05	BCM error
Description	u8_bcmError_t		
Available via	bcm.h		

- en_comm_portocol_t

Name	en_comm_portocol_t		
Type	Enumeration		
Range	UART	0x00	UART
	INVALID_PROTOCOL	0x01	INVALID_PROTOCOL
Description	en_comm_portocol_t		
Available via	bcm.h		

- en_comm_channel_t

Name	en_comm_channel_t		
Type	Enumeration		
Range	UART_0	0x00	UART_0
	INVALID_CHANNEL	0x01	INVALID_CHANNEL
Description	en_comm_channel_t		
Available via	bcm.h		

- en_comm_operation_t

Name	en_comm_operation_t		
Type	Enumeration		
Range	SEND	0x00	SEND
	REC	0x01	REC
Description	en_comm_operation_t		
Available via	bcm.h		

- en_comm_operation_size_t

Name	en_comm_operation_size_t		
Type	Enumeration		
Range	BYTE	0x00	BYTE
	STRING	0x01	STRING
Description	en_comm_operation_t		
Available via	bcm.h		

3.4.1.3 : Services affecting the hardware unit

- BCM_init

Service name	BCM_init	
Syntax	u8_bcmError_t BCM_init(st_bcm_instance_t* st_bcm_instance_t);	
Parameters (in)	st_bcm_instance_t	Pointer to the configuration structure
Return	u8_bcmError_t	BCM_ERROR_OK BCM_ERROR_NOT_OK
Description	This Function Initialize BCM module	

- BCM_deinit

Service name	BCM_deinit	
Syntax	u8_bcmError_t BCM_deinit(st_bcm_instance_t* st_bcm_instance_t);	
Parameters (in)	st_bcm_instance_t	Pointer to the configuration structure
Return	u8_bcmError_t	BCM_ERROR_OK BCM_ERROR_NOT_OK
Description	This Function deInitialize BCM module	

- BCM_send

Service name	BCM_send
Syntax	u8_bcmError_t BCM_send(st_bcm_instance_t*

	st_bcm_instance_t);	
Parameters (in)	st_bcm_instance_t	Pointer to the configuration structure
Return	u8_bcmError_t	BCM_ERROR_OK BCM_ERROR_NOT_OK
Description	This Function send data via BCM module	

- BCM_send_n

Service name	BCM_send_n	
Syntax	u8_bcmError_t BCM_send_n(st_bcm_instance_t* st_bcm_instance_t);	
Parameters (in)	st_bcm_instance_t	Pointer to the configuration structure
Return	u8_bcmError_t	BCM_ERROR_OK BCM_ERROR_NOT_OK
Description	This Function send n of data via BCM module	

- BCM_rec

Service name	BCM_rec	
Syntax	u8_bcmError_t BCM_rec(st_bcm_instance_t* st_bcm_instance_t);	
Parameters (in)	st_bcm_instance_t	Pointer to the configuration structure
Return	u8_bcmError_t	BCM_ERROR_OK BCM_ERROR_NOT_OK
Description	This Function read data via BCM module	

- BCM_rec_n

Service name	BCM_rec_n	
Syntax	u8_bcmError_t BCM_rec_n(st_bcm_instance_t* st_bcm_instance_t);	
Parameters (in)	st_bcm_instance_t	Pointer to the configuration structure
Return	u8_bcmError_t	BCM_ERROR_OK BCM_ERROR_NOT_OK
Description	This Function read n of data via BCM module	

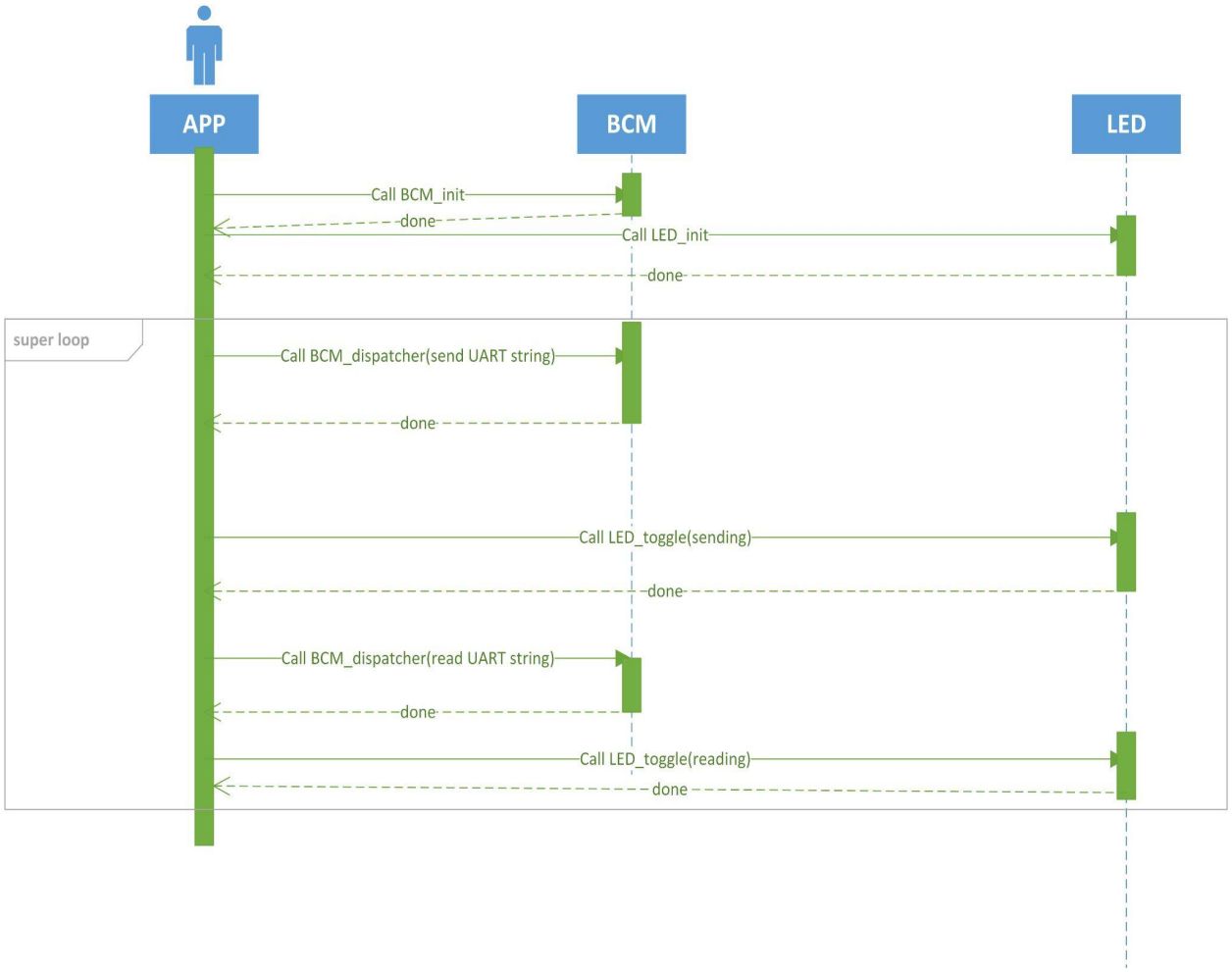
- BCM_dispatcher

Service name	BCM_dispatcher	
Syntax	u8_bcmError_t BCM_dispatcher(st_bcm_instance_t* st_bcm_instance_t);	
Parameters (in)	st_bcm_instance_t	Pointer to the configuration structure
Return	u8_bcmError_t	BCM_ERROR_OK BCM_ERROR_NOT_OK
Description	This Function is the main function in the BCM module	

3.5: APP APIs

3.5.1: APP API:

3.5.1.1 :seq diagram:



3.5.1.2 : Services affecting the hardware unit

- APP_start

Service name	APP_start

Syntax	Void APP_start(void);
Parameters (in)	void
Return	void
Description	This Function starts the app