LED V1 System Design

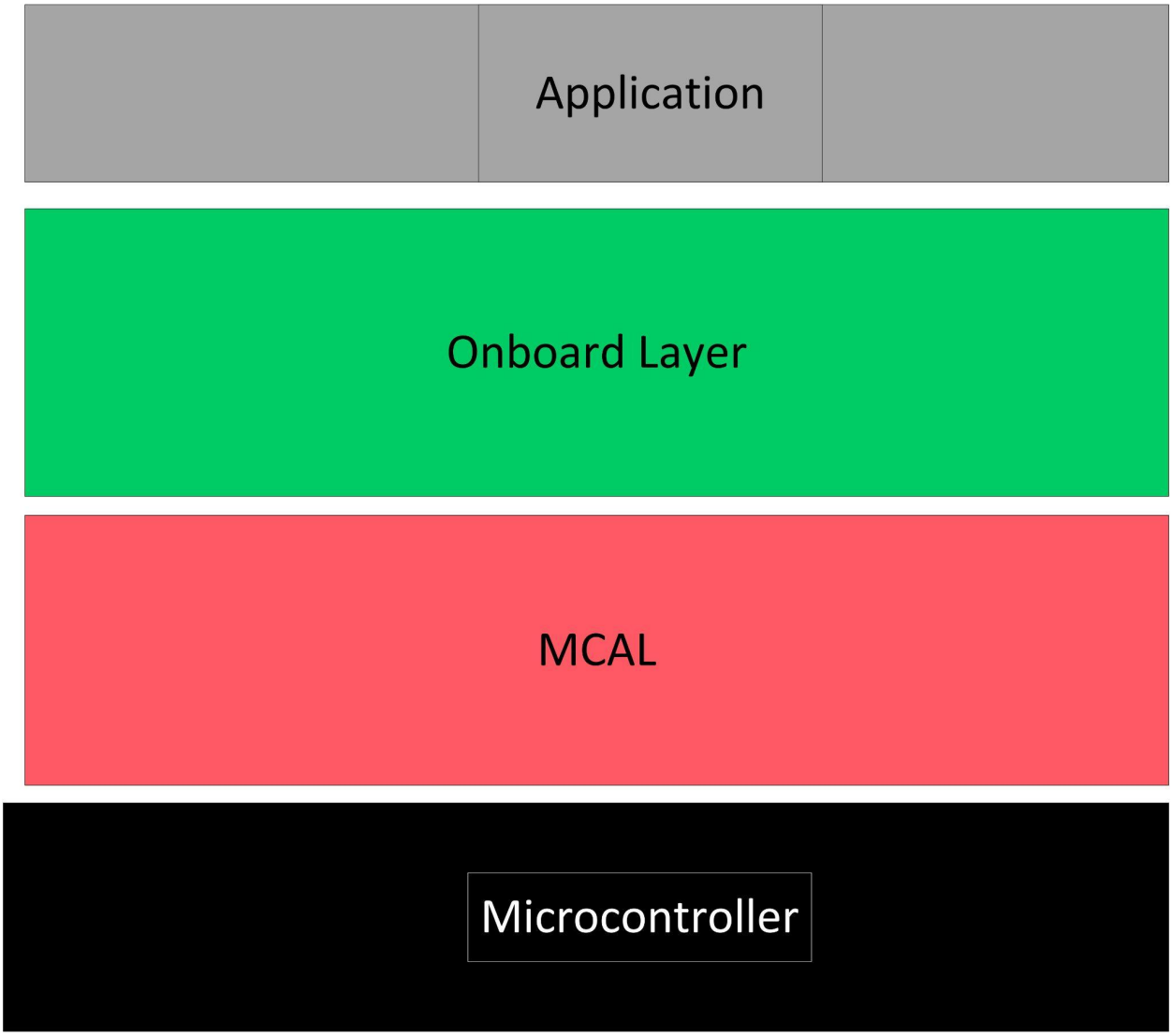
By : Mahmoud Adel Matarawy
Email : mahmoudsarhan02672@gmail.com

Detailed Requirements

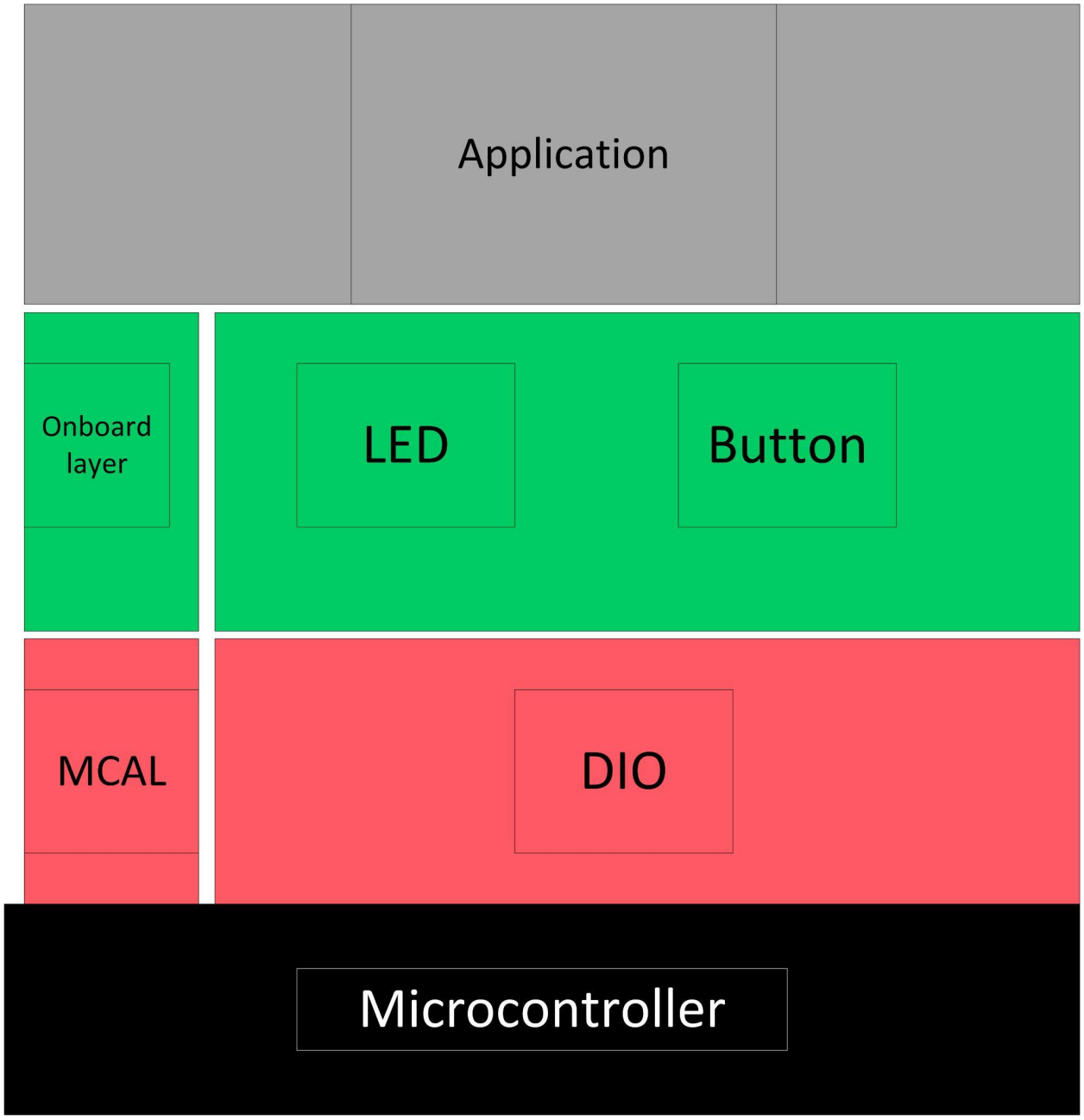
Read System Requirements Specifications

1. *Description*
    1. *Hardware Requirements*
        1. *Four LEDs (LED0, LED1, LED2, LED3)*
        2. One button (BUTTON0)
    2. *Software Requirements*
        1. Initially, all LEDs are OFF
        2. Once BUTTON0 is pressed, LED0 will be ON
        3. Each press further will make another LED is ON
        4. At the fifth press, LED0 will be changed to be OFF
        5. Each press further will make only one LED is OFF
        6. This will be repeated forever
        7. The sequence is described below
            1. Initially (OFF, OFF, OFF, OFF)
            2. Press 1 (ON, OFF, OFF, OFF)
            3. Press 2 (ON, ON, OFF, OFF)
            4. Press 3 (ON, ON, ON, OFF)
            5. Press 4 (ON, ON, ON, ON)
            6. Press 5 (OFF, ON, ON, ON)
            7. Press 6 (OFF, OFF, ON, ON)
            8. Press 7 (OFF, OFF, OFF, ON)
            9. Press 8 (OFF, OFF, OFF, OFF)
            10. Press 9 (ON, OFF, OFF, OFF)

Layered architecture



System modules



APIs

MCAL APIs

DIO API :

Type definitions:

- Dio_ChannelType

Name	Dio_ChannelType
Type	Enumeration
Range	Shall contain all pins ID
Description	Dio_ChannelType
Available via	DIO_Config.h

- Dio_PortType

Name	Dio_PortType
Type	Enumeration
Range	Shall contain all ports ID
Description	Dio_PortType
Available via	DIO_Config.h

- DIO_Errors

Name	DIO_Errors
Type	Enumeration

Range	DIO_E_OK: 0x00 — DIO error OK DIO_InvalidPin: 0x01 — DIO error, invalid pin number.
Description	DIO Errors
Available via	DIO.h

- Dio_LevelType

Name	Dio_LevelType
Type	Enumeration
Range	STD_LOW: 0x00 — Physical state 0V STD_HIGH: 0x01 — Physical state 5V or 3.3V.
Description	Dio_LevelType
Available via	DIO.h

- Dio_DIRType

Name	Dio_DIRType
Type	Enumeration
Range	STD_INPUT: 0x00 — Set pin as input pin STD_OUTPUT: 0x01 — Set pin as output pin
Description	Dio_DIRType
Available via	DIO.h

Services affecting the hardware unit:

- Dio_ReadChannel

Service name	Dio_ReadChannel		
Syntax	DIO_Errors Dio_ReadChannel(Dio_ChannelType ChannelId, Dio_LevelType* level);		
Parameters (in)	ChannelId	Channel ID	
	level	Pointer to store the level	STD_HIGH
			STD_LOW
Return	DIO_Errors	DIO_E_OK	
		DIO_InvalidPin	
Description	This Function gets the level of the pin		

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

- Dio_WriteChannel

Service name	Dio_WriteChannel		
Syntax	DIO_Errors Dio_WriteChannel(Dio_ChannelType ChannelId, Dio_LevelType level);		
Parameters (in)	ChannelId	Channel ID	
	level	Value to be set	STD_HIGH
			STD_LOW
Return	DIO_Errors	DIO_E_OK	
		DIO_InvalidPin	
Description	This Function gets the level of the pin		

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

- Dio_ChannelSetDIR

Service name	Dio_ChannelSetDIR

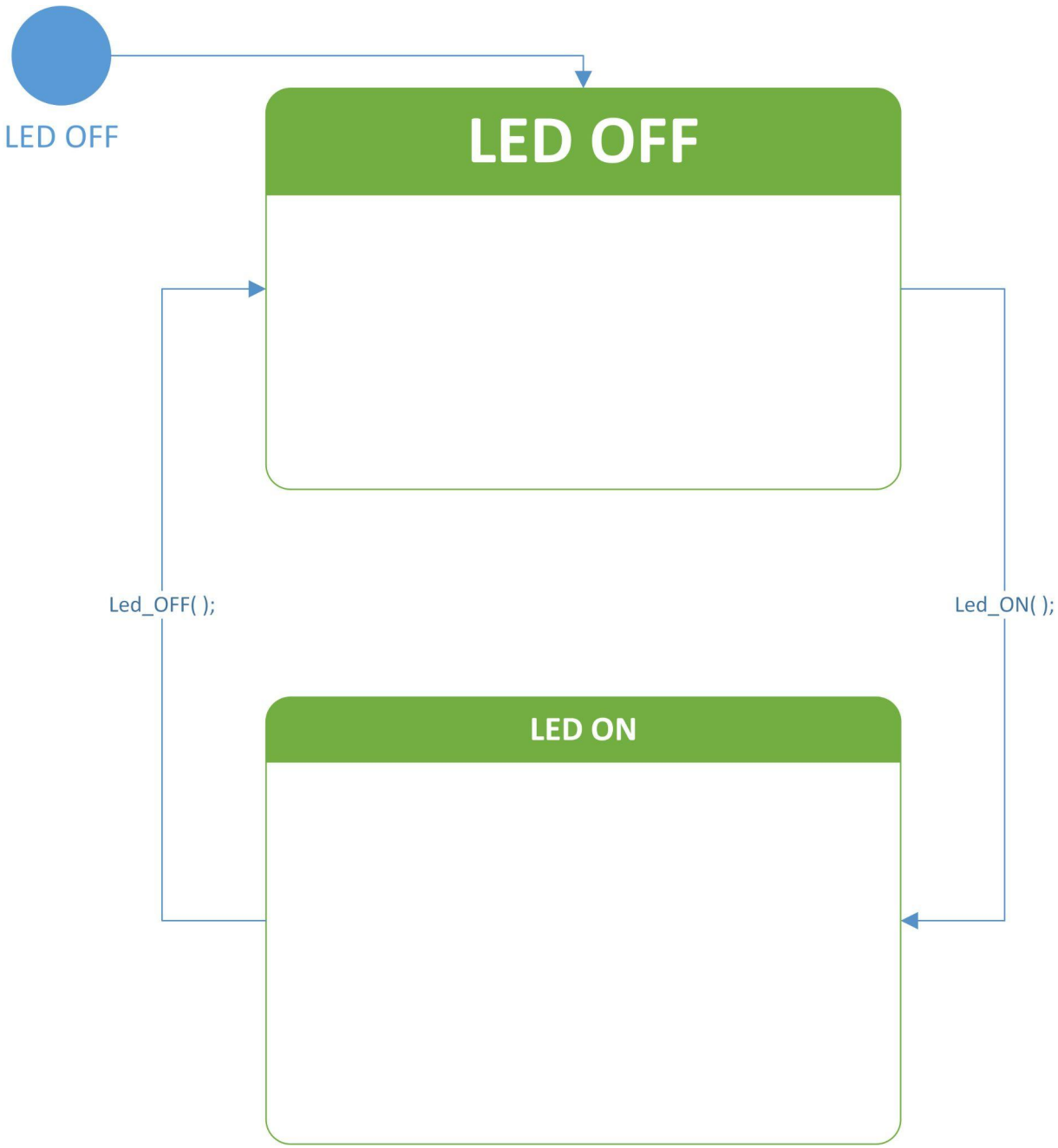
<table>
<tr><td>Syntax</td><td colspan="3">DIO_Errors Dio_ChannelSetDIR(
Dio_ChannelType ChannelId, Dio_DIRType dir
);</td></tr>
<tr><td rowspan="3">Parameters (in)</td><td>ChannelId</td><td colspan="2">Channel ID</td></tr>
<tr><td rowspan="2">dir</td><td rowspan="2">Value to be set</td><td>STD_INPUT</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></td><td>DIO_InvalidPin</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.

Onboard APIs

LED API:

State machine :



Type definitions:

- LED_Config_Type

Name	LED_Config_Type
Type	Structure
Description	This is the type of the external data structure containing the overall configuration data for the LED API
Available via	led_types.h

- LED_STATE_type

Name	LED_STATE_type		
Type	Enumeration		
Range	LED_OFF	0x00	LED OFF STATE
	LED_ON	0x01	LED ON STATE
Description	LED State Enum		
Available via	led.h		

- LED_ID_type

Name	LED_ID_type		
Type	Enumeration		
Range	LED_1	0x01	LED 1
	LED_2	0x02	LED 2
	LED_3	0x03	LED 3
	LED_4	0x04	LED 4
Description	LED ID Enum		
Available via	led.h		

Services affecting the hardware unit:

- led_Init

Service name	led_Init
Syntax	void led_Init(void);
Return	None
Description	This Function Initialize the LED module

- led_OFF

Service name	led_OFF		
Syntax	void led_OFF(LED_ID_type led);		
Parameters (in)	led	LED_1	0x01
		LED_2	0x02
		LED_3	0x03
		LED_4	0x04
Return	None		
Description	This Function Sets a LED OFF.		

- led_ON

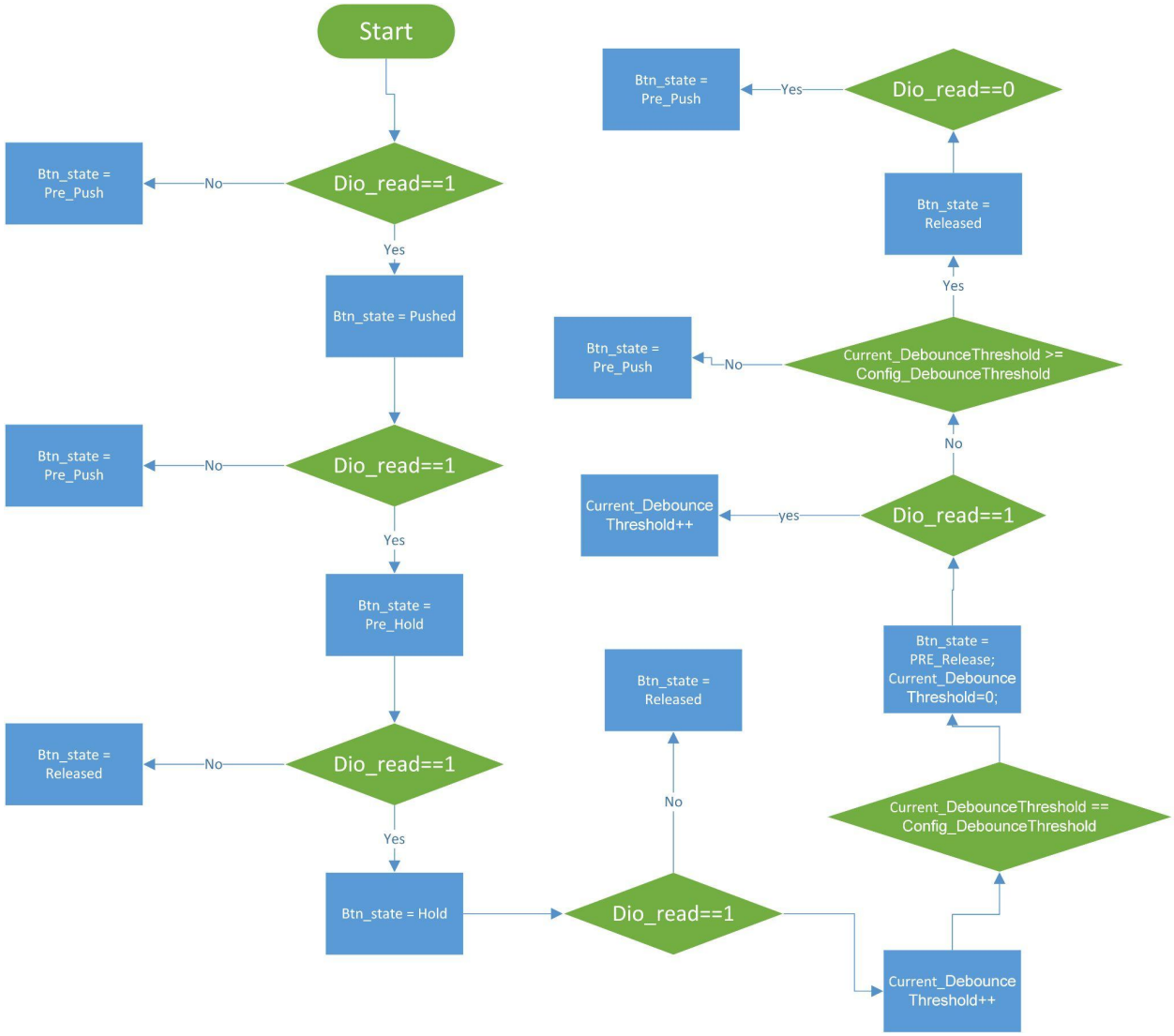
Service name	led_ON		
Syntax	void led_ON(LED_ID_type led);		
Parameters (in)	led	LED_1	0x01
		LED_2	0x02
		LED_3	0x03

		LED_4	0x04
Return	None		
Description	This Function Sets a LED ON.		

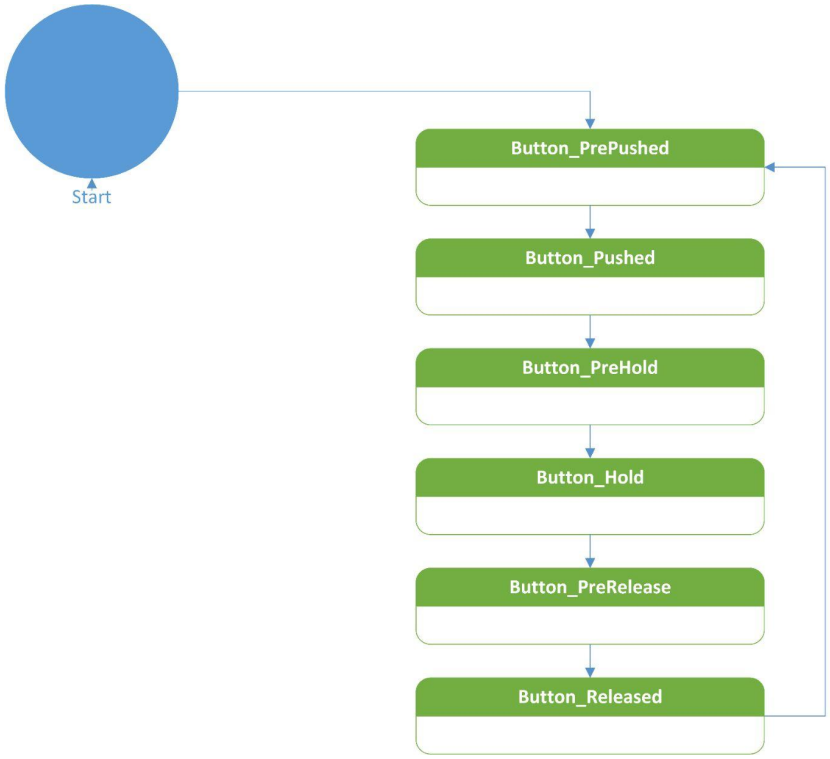
Button API:

Flowchart :

button_Main_Task flowchart



State machine :



Type definitions:

- Button_configType

Name	Button_configType
Type	Structure
Description	This is the type of the external data structure containing the overall configuration data for the Button API
Available via	Button_Types.h

- Button_LevelType

Name	Button_LevelType		
Type	Enumeration		
Range	BT_PUSH_LEVEL	0x00	Push Level
	BT_RELEASE_LEVEL	0x01	Release Level

Description	Button Level Enum
Available via	Button_Types.h

- Button_StateType

Name	Button_StateType		
Type	Enumeration		
Range	BT_PRE_PUSH	0x00	Pre Push Level
	BT_PUSHED	0x01	Pushed Level
	BT_PRE_HOLD	0x02	Pre Hold Level
	BT_HOLD	0x03	Hold Level
	BT_PRE_RELEASE	0x04	Pre Release Level
	BT_RELEASED	0x05	Released Level
	BT_UNDEFINED	0x06	Undefined
Description	Button state Enum		
Available via	Button.h		

- Button_IdType

Name	Button_IdType		
Type	Enumeration		
Range	Button_Start	0x00	Start Button
Description	Button ID Enum		
Available via	Button.h		

Services affecting the hardware unit:

- getButtonState

Service name	getButtonState	
Syntax	Button_StateTyp getButtonState(Button_IdType enmButtonId);	
Parameters (in)	enmButtonId	Start 0x00
Return	Button_StateTyp	BT_PRE_PUSH BT_PUSHED BT_PRE_HOLD BT_HOLD BT_PRE_RELEASE BT_RELEASED BT_UNDEFINED
Description	This Function gets the Button state.	

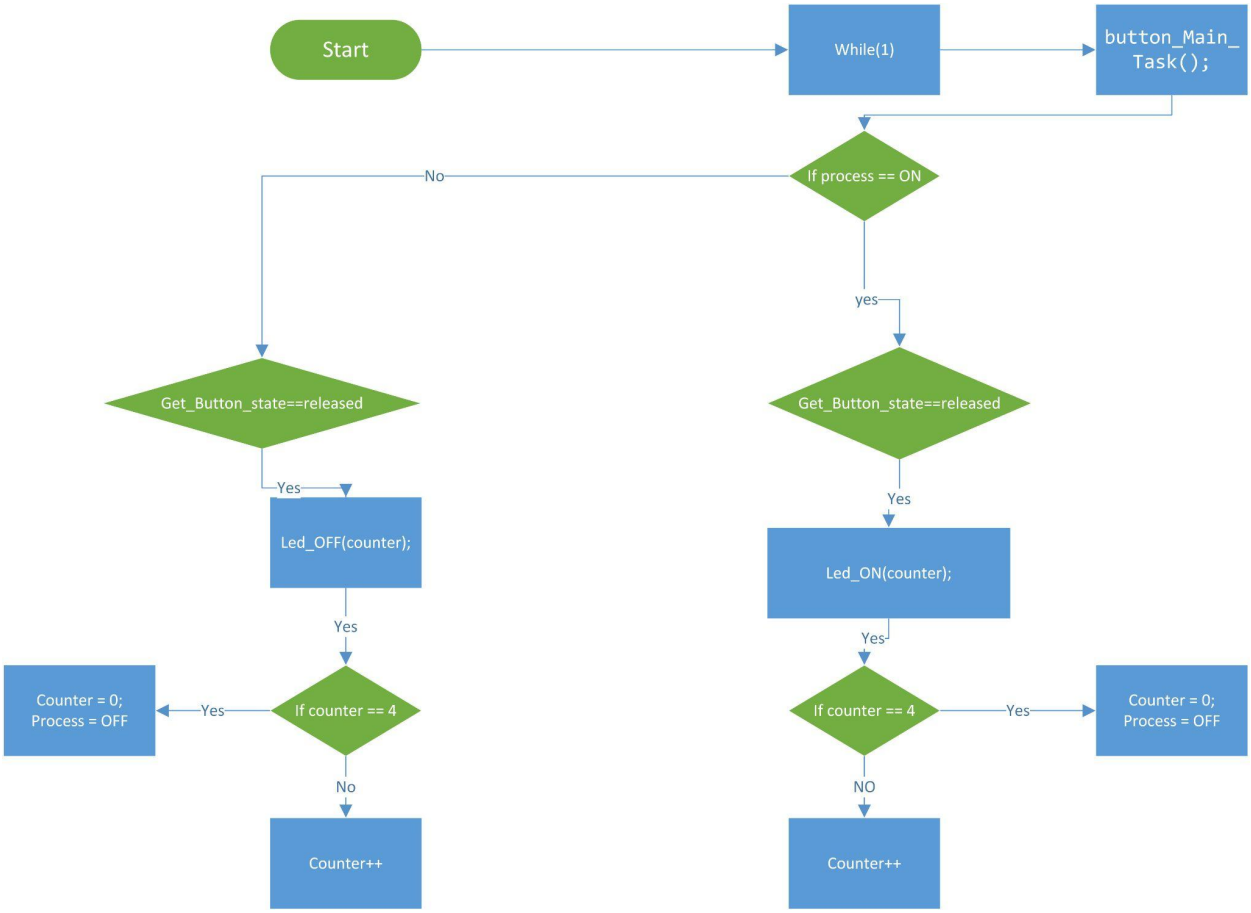
- button_Main_Task

Service name	button_Main_Task
Syntax	void button_Main_Taskt(void);
Parameters (in)	NONE
Return	NONE
Description	This Function update all button states Shall call periodic

App APIs:

App API:

Flowchart :



Services affecting the hardware unit:

- appStart

Service name	appStart
Syntax	void appStart(void);
Parameters (in)	NONE
Return	NONE
Description	This Function Start the application.