BCM Module Design

—

Ebrahem Mostafa
Sprints

Table of content

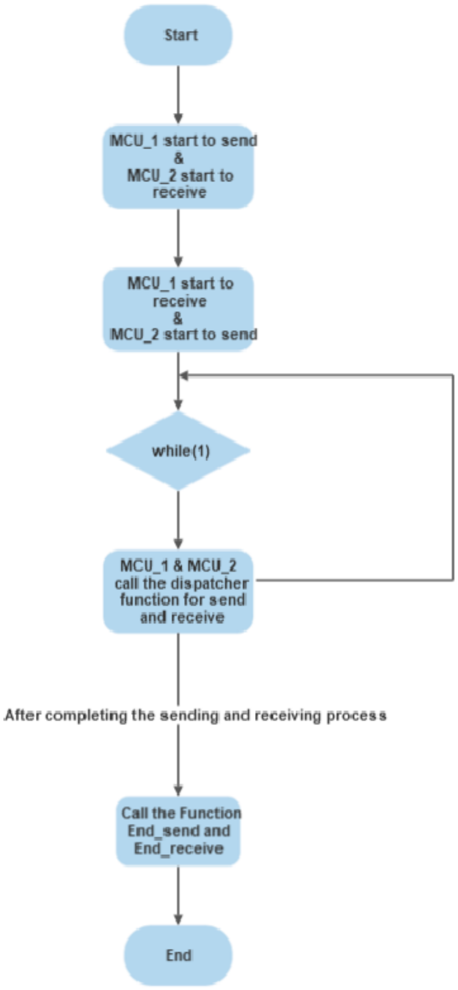
Overview

First of all, The MCU_1 starts to send and receive the same as MCU_2.

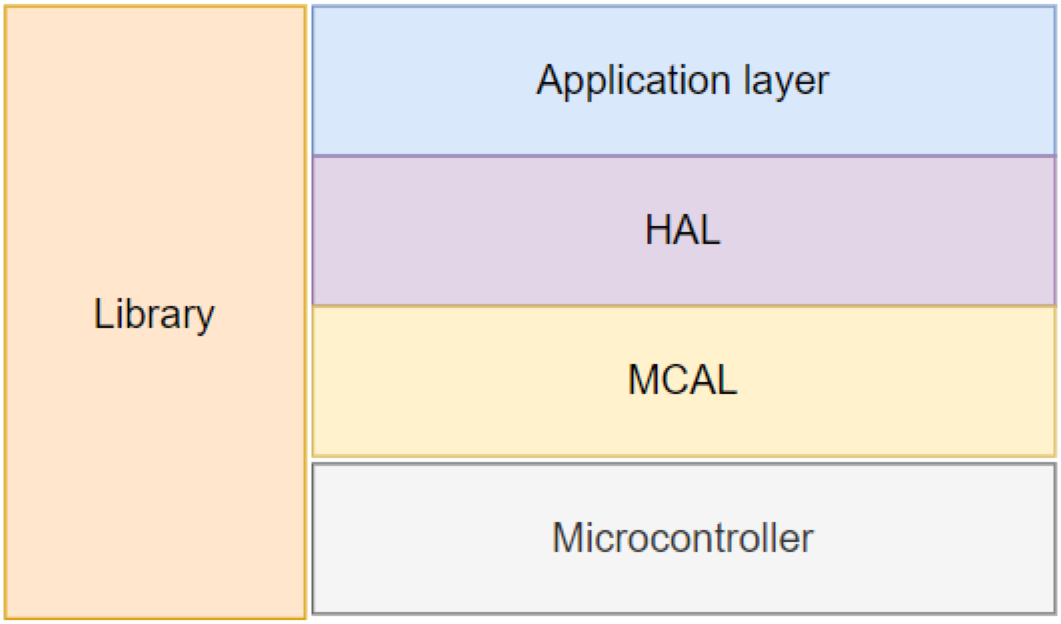
Then, MCU_1 & MCU_2 call the dispatcher function to know if sending and receiving process finish or

not. After that the send_end & receive_end functions will be called.

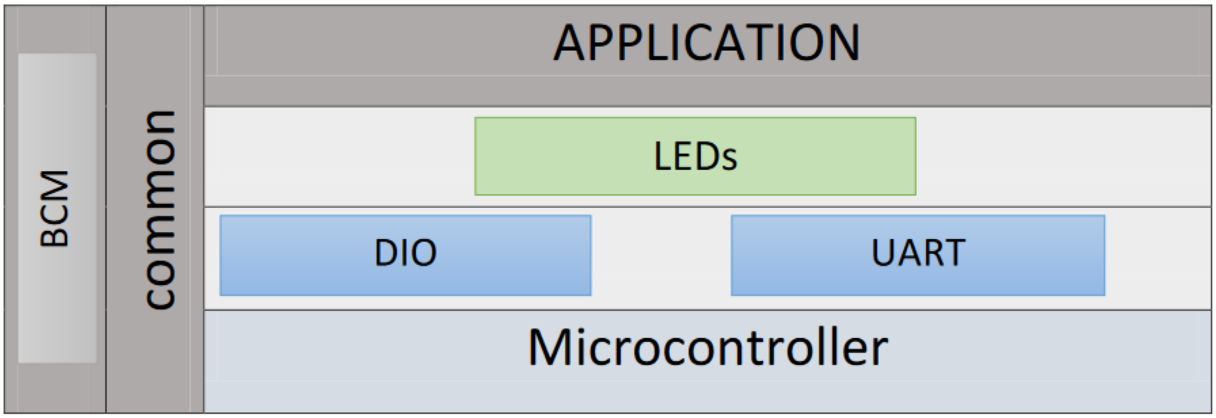
FlowChart


Start
MCU_1 start to send & MCU_2 start to receive
MCU_1 start to receive & MCU_2 start to send
while(1)
MCU_1 & MCU_2 call the dispatcher function for send and receive
After completing the sending and receiving process
Call the Function End_send and End_receive
End

Layered Architecture



System Module



API

- DIO

void DIO_InitPin (PIn_name pin ,PIN_Status status);

void DIO_init (void);

void DIO_WRitePin (PIn_name pin ,Voltage_type s);

Voltage_type DIO_ReadPin(PIn_name pin);

void DIO_WritePort(PORT_Type Port,u8 data);

UART

```
enu_uart_status_t uart_init(void);
void uart_transmit(u8 data);
void uart_transmitNoBlock(u8 data); void
uart_transmitComPlete_InterruptEnable(void);
void uart_transmitComPlete_InterruptDisable(void);
void uart_transmitComPlete_InterruptSetCallback(void(*fptr)(void));
u8 uart_reciever(void);
u8 uart_recieverNoBlock(void);
void uart_recieveComPlete_InterruptEnable(void);
void uart_recieveComPlete_InterruptDisable(void);
void uart_recieveComPlete_InterruptSetCallback(void(*fptr)(void));
```

LED

```
 void LED_init(u8 led);
void LED_ON (u8 LED );
void LED_OFF (u8 LED );
void LED_Toggle (u8 LED );
```

Application

```
void app_init();

void app_start();
```

- ### BCM

  enu_system_status_t bcm_init(enu_protocol_t protocol);

  enu_system_status_t bcm_deinit(enu_protocol_t protocol);

  void bcm_send(u8 data); //Blocking Function

  void bcm_send_Blocking(u8*str);// Blocking function

  void bcm_send_Non_Blocking(u8*str,u8 length, u8 start); //Non Blocking function

  void uart_bcm_send_dispatcher(void);

  void bcm_send_End_Setcallback(void (*fptr)(void));

  void bcm_receive(u8*data); //Blocking Function

  void bcm_receive_Blocking(u8*str); //Blocking Function

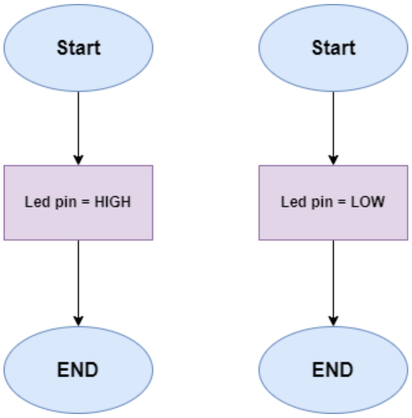
  void bcm_receive_Non_Blocking(u8*str); //Non Blocking function

  void bcm_receive_End_Setcallback(void (*fptr)(void));

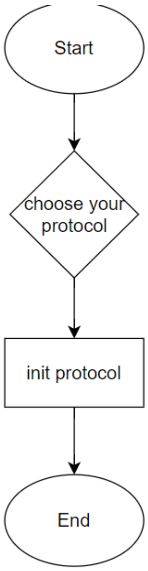
  void uart_bcm_recieve_dispatcher(void);

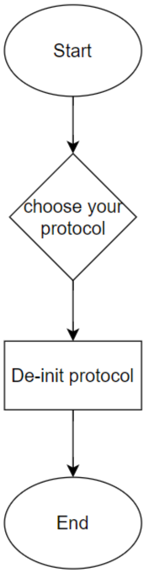
API Flowchart

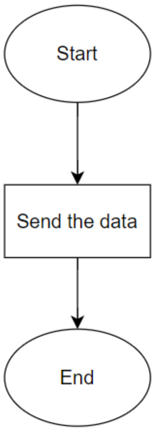
- ### Led

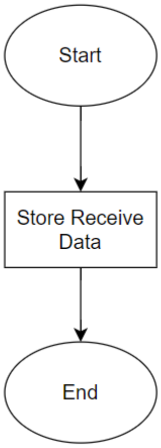


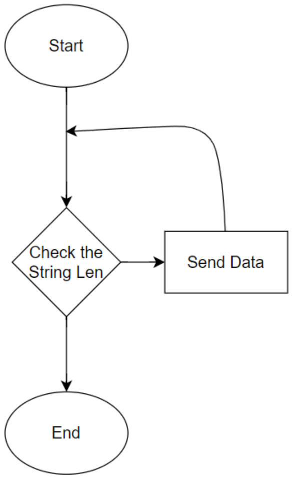
- BCM

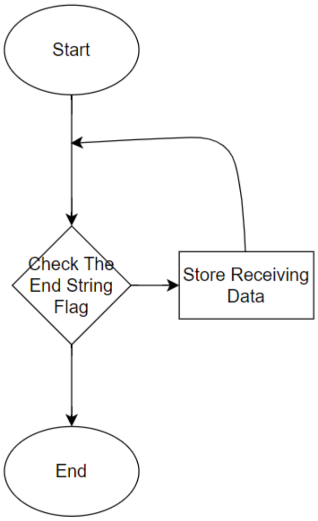

Start
choose your protocol
init protocol
End

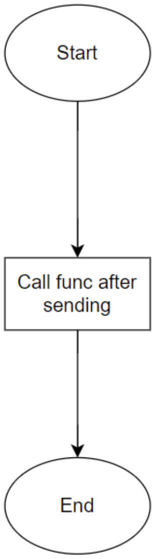

Start
choose your protocol
De-init protocol
End


Start
Send the data
End

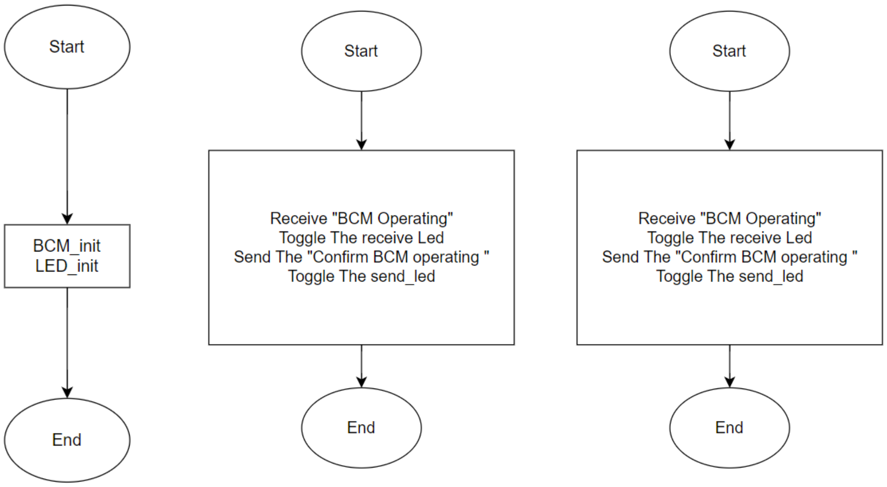

Start
Store Receive Data
End


Start
Check the String Len
Send Data
End


Start
Check The End String Flag
Store Receiving Data
End


Start
Call func after sending
End

- app


Start
BCM_init
LED_init
End
Start
Receive "BCM Operating"
Toggle The receive Led
Send The "Confirm BCM operating "
Toggle The send_led
End
Start
Receive "BCM Operating"
Toggle The receive Led
Send The "Confirm BCM operating "
Toggle The send_led
End