Sprints

Description:

You are required to design a BCM (Basic Communication Manager). This module has a capability to work with different serial communication protocols using ISR with the highest possible throughput.

Detailed Requirements

1. **Read System Requirement Specifications**
    1. The BCM has the capability to send and receive any data with maximum length of **65535** bytes (Maximum of unsigned two bytes variable).
    2. It can use any communication protocol with the support of **Send**, **Receive** or **both**.
    3. Implement **bcm_Init** use the below table. This function will initialize the corresponding serial communication protocol

Function Name	bcm_init
Syntax	enu_system_status_t bcm_init (str_bcm_instance_t* ptr_str_bcm_instance);
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in):	ptr_str_bcm_instance: Address of the BCM Instance
Parameters (out):	None
Parameters (in, out):	None
Return:	You have to fill it

    4. Implement **bcm_deinit** use the below table. This function will uninitialize the corresponding BCM instance, (**instance: is the communication channel**)

Function Name	bcm_deinit
Syntax	enu_system_status_t bcm_deinit (str_bcm_instance_t* ptr_str_bcm_instance);
Sync/Async	Synchronous
Reentrancy	Non Reentrant
Parameters (in):	ptr_str_bcm_instance: Address of the BCM Instance
Parameters (out):	None
Parameters (in,out):	None
Return:	You have to fill it

    5. Implement **bcm_send** that will send only **1 byte of data** over a specific BCM instance (please create a table for this function as the previous functions).

6. Implement **bcm_send_n** will send **more than one byte with a length n** over a specific BCM instance (please create a table for this function as the previous functions).
7. Implement **bcm_dispatcher** will execute the periodic actions and notifies the user with the needed events over a specific BCM instance (please create a table for this function as the previous functions).

2. **Module testing**
    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.
3. **Prepare your design**
    1. Create a PDF file with the name **Basic Communication Manager Design**
    2. The design document should contain the below fields
        1. Cover Page
        2. Table of content
        3. Project introduction
        4. High Level Design
            1. Layered architecture
            2. Modules Descriptions
            3. Drivers' documentation
            4. UML
            5. Sequence diagram
        5. Low Level Design
            1. Provide the flowchart for each function in each module
            2. Pre-compiling configurations for each module
            3. Linking configurations for each module

Delivery

1. Deliver the Design Document
2. English Video recording 5 minutes maximum discuss your design