Basic Communication Manager

Design

By

Ahmed Atef

Contents

1.   Project Introduction

This project is aiming to deliver a BCM -Basic Communication Manager- which will manage the data which the user wants to transmit/receive to make a specific task. I implemented the project to send data using UART protocol which is up to 512 byte and it can be from 0 to 255
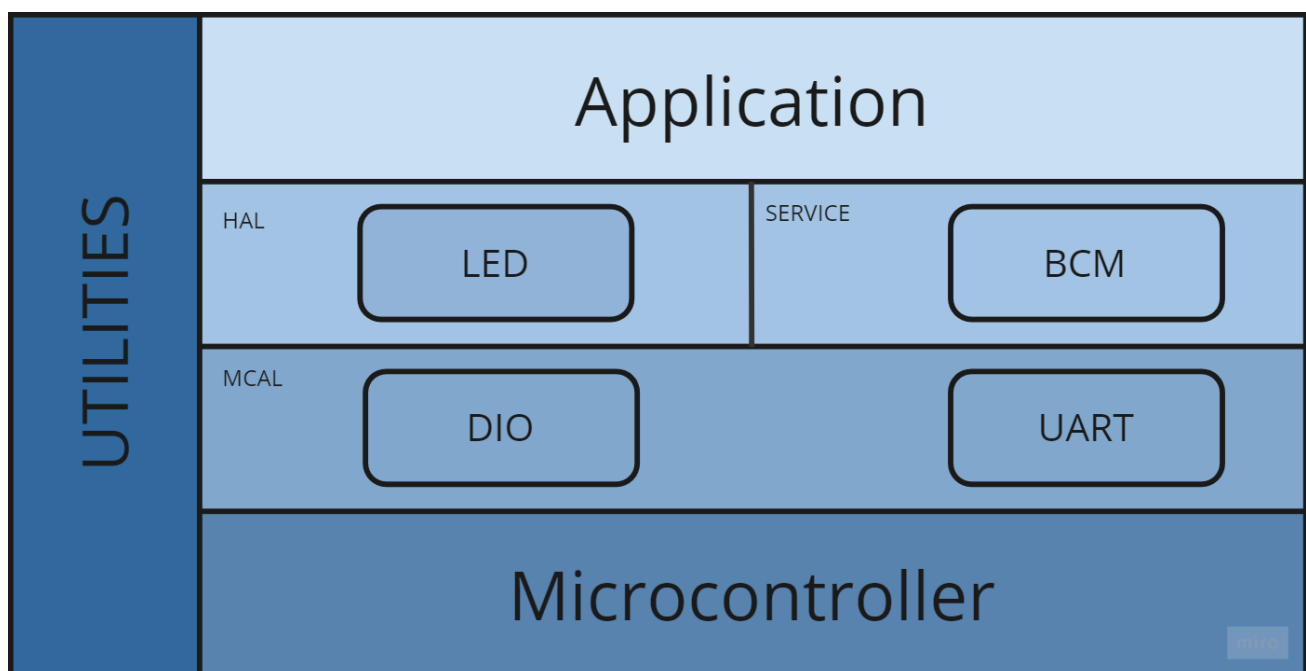
1.1.   Project Components

- 2 ATmega32 microcontroller
- 2 LEDS for each microcontroller

2.   High Level Design

2.1.   System Architecture

2.1.1.   Layered Architecture

2.2.   BCM Data Buffer

BCM DATA BUFFER

dataBus_0 1 byte	dataSize_0 2 bytes	data_0 n of bytes	dataBus_1 1 byte	dataSize_1 2 bytes	data_1 n of bytes	dataBus_# 1 byte	dataSize_# 2 bytes	data_# n of bytes

2.3.   Modules Description

2.3.1.   DIO (Digital Input/Output) Module

The DIO driver is responsible for reading input signals from the system's sensors (such as buttons) and driving output signals to the system's actuators (such as LEDs). It provides a set of APIs to configure the direction and mode of each pin (input/output, pull-up/down resistor), read the state of an input pin, and set the state of an output pin.

2.3.2.   UART

The UART driver is responsible for initializing UART channel to configured specifications, it is used to send data through it and receive data too. All of its tasks are non-blocking.

2.3.3.   LED

The LED driver is used to initialize LEDs used as output and control them as it can turn them on or off or toggle them.

2.3.4.   BCM

The BCM driver is used to initialize communication protocols used to send/receive specific data types, its responsible to manage the communication process.

2.4.   Drivers' Documentation

2.4.1.   MCAL Drivers' Functions

2.4.1.1.   DIO Driver

DIO_init

Syntax	DIO_init(uint8_t uint8_portNumber, uint8_t uint8_pinNumber, uint8_t uint8_direction)	
Description	Initializes DIO pins' direction, output current.	
Sync\Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	uint8_portNumber	port number used
	uint8_pinNumber	pin number used
	uint8_direction	direction of the pin
Parameters (out)	None	
Return value	enu_DIO_status_t	DIO_OK
		WRONG_PIN_NUMBER
		WRONG_PORT_NUMBER
		WRONG_DIRECTION

DIO_write

Syntax	DIO_write(uint8_t uint8_portNumber, uint8_t uint8_pinNumber, uint8_t uint8_value)	
Description	Write on DIO pins' a specific output High or Low.	
Sync\Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	uint8_portNumber	port number used
	uint8_pinNumber	pin number used
	uint8_value	level of the pin
Parameters (out)	None	
Return value	enu_DIO_status_t	DIO_OK
		WRONG_PIN_NUMBER
		WRONG_PORT_NUMBER
		WRONG_VALUE

DIO_toggle

Syntax	DIO_toggle(uint8_t uint8_portNumber, uint8_t uint8_pinNumber)
Description	Toggle the output of a specific pin.
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	uint8_portNumber port number used uint8_pinNumber pin number used
Parameters (out)	None

Return value	enu_DIO_status_t DIO_OK
	                                        WRONG_PIN_NUMBER
	                                        WRONG_PORT_NUMBER

DIO_read

Syntax	DIO_read(uint8_t uint8_portNumber, uint8_t uint8_pinNumber, uint8_t *uint8_value)
Description	Read input from a pin and send it back in a pointer to uint8_t
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	uint8_portNumber port number used uint8_pinNumber pin number used
Parameters (out)	uint8_value input value will be returned in that parameter

Return value	enu_DIO_status_t DIO_OK
	                                          WRONG_PIN_NUMBER
	                                        WRONG_PORT_NUMBER

DIO_pinPullUp

Syntax	DIO_pinPullUp(uint8_t uint8_portNumber, uint8_t uint8_pinNumber, uint8_t uint8_pullUpState)
Description	Disables/enables a pull up resistor to a specific input pin
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	uint8_portNumber port number used uint8_pinNumber pin number used uint8_pullUpState pullup state
Parameters (out)	None

Return value	enu_DIO_status_t DIO_OK
	                                          WRONG_PIN_NUMBER
	                                        WRONG_PORT_NUMBER
	                                        WRONG_VALUE

2.4.1.2. UART Driver

UART_init

Syntax	UART_init (void)	
Description	Initializes UART pins' direction, and specifications	
Sync\Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (out)	None	
Return value	enu_UART_status_t	UART_OK = 0, UART_NOK

UART_deinit

Syntax	UART_deinit (void)	
Description	Deinitializes UART	
Sync\Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	None	
Parameters (out)	None	
Return value	enu_UART_status_t	UART_OK = 0, UART_NOK

UART_sendByte

Syntax	UART_sendByte (uint8_t byte)	
Description	Sending one byte through UART channel	
Sync\Async	Asynchronous	
Reentrancy	Reentrant	
Parameters (in)	uint8_byte	
Parameters (out)	None	
Return value	enu_UART_status_t	UART_OK = 0, UART_SENDING = 1, UART_NOK

UART_setCallBack

Syntax	UART_setCallBack (void (*ptr_func)(void))	
Description	Set callback function	
Sync\Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	void (*ptr_func)(void)	
Parameters (out)	None	
Return value	enu_UART_status_t	UART_OK = 0, UART_NOK

UART_receiveData

Syntax	UART_receiveData(uint8_t** ptr_uint8_receivedData, uint16_t* uint16_dataSize)
Description	Receives data buffer
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (out)	ptr_ptr_uint8_receivedData ptr_uint16_dataSize
Return value	enu_UART_status_t UART_OK = 0, UART_NOK

UART_isEmpty

Syntax	UART_isEmpty (void)
Description	Checks if data register is empty
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (out)	ptr_uint8_receivedData ptr_uint16_dataSize
Return value	enu_UART_bufferStatus_t UART_BUFFER_NEMPTY UART_BUFFER_EMPTY

2.4.2.   HAL Drivers' Functions

2.4.2.1.   LED

LED_init

Syntax	LED_init(void)
Description	Initializes LED pins' direction as output
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	None
Parameters (out)	None
Return value	enu_LED_status_t LED_OK= 0, LED_WRONG_LED_PORT LED_WRONG_LED_PIN

LED_on

Syntax	LED_on(uint8_t uint8_ledID)
Description	Turns on a specific LED
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	uint8_ledID
Parameters (out)	None
Return value	enu_LED_status_t LED_OK= 0, LED_WRONG_LED_PORT LED_WRONG_LED_PIN

LED_off

Syntax	LED_off(uint8_t uint8_ledID)
Description	Turns off a specific LED
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	uint8_ledID
Parameters (out)	None
Return value	enu_LED_status_t LED_OK= 0, LED_WRONG_LED_PORT LED_WRONG_LED_PIN

LED_toggle

Syntax	LED_toggle(uint8_t uint8_ledID)	
Description	Toggles a specific LED	
Sync\Async	Synchronous	
Reentrancy	Reentrant	
Parameters (in)	uint8_ledID	
Parameters (out)	None	
Return value	enu_LED_status_t	LED_OK= 0, LED_WRONG_LED_PORT LED_WRONG_LED_PIN

2.4.3.   SERVER Drivers' functions

2.4.3.1.   BCM

BCM_init

Syntax	BCM_init (str_BCM_instance_t* ptr_str_BCM_instance)
Description	Initializes a specific communication protocol
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	ptr_str_BCM_instance
Parameters (out)	None
Return value	enu_BCM_status_t BCM_OK = 0, BCM_NOK

BCM_deinit

Syntax	BCM_deinit (str_BCM_instance_t* ptr_str_BCM_instance)
Description	Deinitializes a specific communication protocol
Sync\Async	Synchronous
Reentrancy	Reentrant
Parameters (in)	ptr_str_BCM_instance
Parameters (out)	None
Return value	enu_BCM_status_t BCM_OK = 0, BCM_NOK,

BCM_dispatcher

Syntax	BCM_dispatcher (void)
Description	Dequeue the data from data buffer and send it through a specific communication protocol
Sync\Async	Asynchronous
Reentrancy	Reentrant
Parameters (in)	ptr_str_BCM_instance
Parameters (out)	None

Return value	enu_BCM_status_t BCM_OK = 0,
	                                    BCM_NOK,
	                                    BCM_EMPTY

3. Low Level Design

BCM_RxDispatcher

start

error check

if gu8RX_Flag isHIGH ──No──→

switch bcm_state

Yes

IDLE ──Yes──→ change the state depend on isr flag

No

TXCMD ──Yes──→ Check if there cmd to recieve data

No

TXSIZE ──Yes──→ recive and store the size

No

TXDATA ──Yes──→ recive data from data buffar

No

TXCS ──Yes──→ do the notfy function that u need after coomplet reciving

No

FINISHED ──Yes──→ finsh swich with make all fages in default values

No

default case ──→ end switch

end