UART CDD

for

LCD\_LED\_UART

**Prepared by: Moamen Ahmed**

**Mar 27, 2020**

**Contents**

[**1.**](#_gjdgxs) **Global Variables** 1

[**2.**](#_30j0zll) **Configurations** 1

[**3.**](#_1fob9te) **UART APIs** 2

# **Global Variables**

**static u8 TX\_Buffer[32]**

**transmit message buffer.**

**static u16 TX\_Buffer\_Len**

**transmit message length.**

**static u8 RX\_Buffer[32]**

**receive message buffer.**

**static u16 RX\_Buffer\_Len**

**receive message length.**

**rxCbf\_t txCallback**

**transmit callback function**

**rxCbf\_t rxCallback**

**recieve callback function**

# **Configurations**

**N/A**

1. **GPIO APIs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | UART | | |
| **API Name** | STD\_ERROR UART\_Init(void) | | |
| **Return type** | It’s u8 Error\_Status, it returns OK or NOT\_OK   |  |  | | --- | --- | | OK | 0 | | NOT\_OK | 1 | | | |
| **Input signal** | * N/A | Output signal | N/A |
| **Description** | The functionality of this API is to configure UART to settings defined in UART\_config.h | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | UART | | |
| **API Name** | void UART\_Send(u8 \* buffer, u16 len) | | |
| **Return type** | N/A | | |
| **Input signal** | buffer:   * Type: u8\* * Description: message buffer to send   len:   * Type: u16 * Description: length of message | Output signal | N/A |
| **Description** | Send a message through UART | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | UART | | |
| **API Name** | void UART\_Receive(u8\* buffer, u16\* len) | | |
| **Return type** | N/A | | |
| **Input signal** | buffer:   * Type: u8\* * Description: pointer to receive message   len:   * Type: u16\* * Description: pointer to receive message length | Output signal | buffer:  message  len:  message length |
| **Description** | Receive message over UART | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | UART | | |
| **API Name** | STD\_ERROR UART\_Configure(u32 baudrate, u8 stopBits, u32 parity) | | |
| **Return type** | It’s u8 Error\_Status, it returns OK or NOT\_OK   |  |  | | --- | --- | | OK | 0 | | NOT\_OK | 1 | | | |
| **Input signal** | baudrate:   * Type: u32 * Description: set baudrate   stopBits:   * Type: u8 * Description: stop bits settings   parity:   * Type: u32 * Description: parity bits settings | Output signal | N/A |
| **Description** | Configure UART settings | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | UART | | |
| **API Name** | STD\_ERROR UART\_SetTxCallbackFnc(txCbf\_t txcbf) | | |
| **Return type** | It’s u8 Error\_Status, it returns OK or NOT\_OK   |  |  | | --- | --- | | OK | 0 | | NOT\_OK | 1 | | | |
| **Input signal** | txcbf:   * Type: txCbf\_t * Description: pointer to callback function | Output signal | N/A |
| **Description** | Set callback function for transmitter | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | UART | | |
| **API Name** | STD\_ERROR UART\_SetRxCallbackFnc(rxCbf\_t rxcbf) | | |
| **Return type** | It’s u8 Error\_Status, it returns OK or NOT\_OK   |  |  | | --- | --- | | OK | 0 | | NOT\_OK | 1 | | | |
| **Input signal** | rxcbf:   * Type: rxCbf\_t * Description: pointer to callback function | Output signal | N/A |
| **Description** | Set callback function for receiver | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | UART | | |
| **API Name** | void UART\_DefaultTxCallback(void) | | |
| **Return type** | N/A | | |
| **Input signal** | N/A | Output signal | N/A |
| **Description** | Default callback function for transmitter | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | UART | | |
| **API Name** | void UART\_DefaultRxCallback(void) | | |
| **Return type** | N/A | | |
| **Input signal** | N/A | Output signal | N/A |
| **Description** | Default callback function for receiver | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | UART | | |
| **API Name** | void USART1\_IRQHandler(void) | | |
| **Return type** | N/A | | |
| **Input signal** | N/A | Output signal | N/A |
| **Description** | USART1 default interrupt handler | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | UART | | |
| **API Name** | void UART\_TransmitStatus(u8 \* stat) | | |
| **Return type** | N/A | | |
| **Input signal** | stat:   * Type: u8 \* * Description: bus status | Output signal | stat:  bus status |
| **Description** | Check if transmission is finished | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | UART | | |
| **API Name** | void UART\_RecieveStatus(u8 \* stat) | | |
| **Return type** | N/A | | |
| **Input signal** | stat:   * Type: u8 \* * Description: bus status | Output signal | stat:  bus status |
| **Description** | Check if receiveis finished | | |
| **Type (Public/Private)** | Public | | |