HLCD CDD

for

LCD\_LED\_UART

Prepared by: Moustafa Ghareeb

Mar 27, 2020

**Contents**

[**1.** **Global Variables** 1](#_Toc36231083)

[**2.** **Configurations** 1](#_Toc36231088)

[**3.** **HLCD APIs** 3](#_Toc36231089)

# **Global Variables**

### HLCD\_state: Holds the current state of the LCD.

### HLCD\_busy: Holds the Busy or Idle state for the LCD.

### HLCD\_data\_cmd: Holds the data or command to be applied to the LCD.

### dataCounter: Holds the size of the data to be written on the LCD.

# **Configurations**

|  |  |
| --- | --- |
| **Name** | HLCD\_E\_PIN  HLCD\_RS\_PIN  HLCD\_RW\_PIN  HLCD\_D0\_PIN  HLCD\_D1\_PIN  HLCD\_D2\_PIN  HLCD\_D3\_PIN  HLCD\_D4\_PIN  HLCD\_D5\_PIN  HLCD\_D6\_PIN  HLCD\_D7\_PIN |
| **Description** | To configure the LCDs pins. |
| **Element** | PIN0 - PIN1 - PIN2 - PIN3 - PIN4 - PIN5 - PIN6 - PIN7 - PIN8  PIN9 - PIN10 - PIN11 - PIN12 - PIN13 - PIN14 - PIN15 |
| **Range** | 1 – 2 – 4 – 8 – 16 – 32 – 64 – 128 – 256 – 512 – 1024 - 2048 – 4096 – 8192 - 16384 - 32768 |

|  |  |
| --- | --- |
| **Name** | HLCD\_E\_PORT  HLCD\_RS\_PORT  HLCD\_RW\_PORT  HLCD\_D0\_PORT  HLCD\_D1\_PORT  HLCD\_D2\_PORT  HLCD\_D3\_PORT  HLCD\_D4\_PORT  HLCD\_D5\_PORT  HLCD\_D6\_PORT  HLCD\_D7\_PORT |
| **Description** | To configure the LCDs ports. |
| **Element** | PORTA  PORTB  PORTC |
| **Range** | 0 – 1 – 2 |

1. **HLCD APIs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | HLCD | | |
| **API Name** | STD\_ERROR HLCD\_u8Init(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 initialize the LCD pins with default configurations. | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | HLCD | | |
| **API Name** | STD\_ERROR HLCD\_u8WriteDataRequest(u8\* Copy\_u8Data, u8 Copy\_u8Size) | | |
| **Return type** | It’s u8 Error\_Status, it returns OK or NOT\_OK   |  |  | | --- | --- | | OK | 0 | | NOT\_OK | 1 | | | |
| **Input signal** | Copy\_u8Data :   * Type: u8\* * Description: Data to be written on the LCD   Copy\_ u8Size:   * Type: u8 * Description: The size of the data | Output signal | N/A |
| **Description** | The functionality of this API is to save the data to be written on the LCD in case of the LCD being idle. | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | HLCD | | |
| **API Name** | STD\_ERROR HLCD\_u8WriteCmdRequest(u8 Copy\_u8Cmd) | | |
| **Return type** | It’s u8 Error\_Status, it returns OK or NOT\_OK   |  |  | | --- | --- | | OK | 0 | | NOT\_OK | 1 | | | |
| **Input signal** | Copy\_u8Cmd:   * Type: u8 * Description: Data to be applied on the LCD | **Output signal** | N/A |
| **Description** | The functionality of this API is to save the command to be applied on the LCD in case of the LCD being idle. | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | HLCD | | |
| **API Name** | STD\_ERROR HLCD\_u8CursorPosition(u8 Copy\_u8Row, u8 Copy\_u8Column) | | |
| **Return type** | It’s u8 Error\_Status, it returns OK or NOT\_OK   |  |  | | --- | --- | | OK | 0 | | NOT\_OK | 1 | | | |
| **Input signal** | Copy\_u8Row:   * Type: u8 * Description: Desired row number for the cursor [0-1]   Copy\_u8Column:   * Type: u8 * Description: Desired col number for the cursor [0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15] | Output signal | N/A |
| **Description** | The functionality of this API is to change the cursor position. | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | HLCD | | |
| **API Name** | void HLCD\_voidRunnable(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 do the initialization sequence required to start the LCD then check the local buffer for a data or a command to apply it on the LCD. | | |
| **Type (Public/Private)** | Public | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | HLCD | | |
| **API Name** | void HLCD\_voidWriteData(u8 Copy\_u8Data) | | |
| **Return type** | N/A | | |
| **Input signal** | Copy\_u8Data:   * Type: u8 * Description: A character to be written on the LCD | Output signal | N/A |
| **Description** | The functionality of this API is to write one character on the LCD. | | |
| **Type (Public/Private)** | Private | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | HLCD | | |
| **API Name** | void CLCD\_voidWriteCmd(u8 Copy\_u8Command) | | |
| **Return type** | N/A | | |
| **Input signal** | Copy\_u8Command:   * Type: u8   Description: A command to be applied on the LCD | Output signal | N/A |
| **Description** | The functionality of this API is to apply a command on the LCD. | | |
| **Type (Public/Private)** | Private | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Component Name** | HLCD | | |
| **API Name** | void HLCD\_voidWriteDataCmd(u8 Copy\_u8DataCmd) | | |
| **Return type** | N/A | | |
| **Input signal** | Copy\_u8DataCmd:   * Type: u8   Description: A command or a data character to be applied on the LCD. | Output signal | N/A |
| **Description** | The functionality of this API is to assign each LCD data pin to the required value. | | |
| **Type (Public/Private)** | Private | | |