Character LCD

|  |  |
| --- | --- |
| **Document Name** |  |
| **Version** |  |
| **Author** | Norhan Osama |
| **Status** |  |
| **Date** | 4/1/2016 |

Document History

| Version | Date | Author | Changes |
| --- | --- | --- | --- |
| 1.00 | 01/04/2016 | Norhan Osama | Initial |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of content

[1 Introduction 4](#_Toc447289905)

[1.1 Goals and Objectives 4](#_Toc447289906)

[1.2 Component APIs and variables 4](#_Toc447289907)

[2 Design Constraints 6](#_Toc447289908)

[2.1 Constraints on Initialization 6](#_Toc447289909)

[2.2 Constraints on Inputs 6](#_Toc447289910)

[3 Configuration 7](#_Toc447289911)

[3.1 File Description 7](#_Toc447289912)

# Introduction

This document is created to demonstrate the character LCD component and its used APIs also to demonstrate the dependant drivers for this module.

## Goals and Objectives

To Provide demonstration of the used APIs and components to write any string on the LCD, controlling where to display the character on the LCD.

Display any kind of characters by creating custom characters on the LCD internal CGRAM, so basically we can write any alphabetical characters in any language.

## Component APIs and variables

|  |  |
| --- | --- |
| **Interface** | **Description** |
| *LCD driver* | *The driver is to write characters on the LCD and interface with the DIO driver.* |
| *DIO Driver* | *It is required to interface with microcontroller through DIO peripheral.* |

Table 1: Interfaces description

|  |  |  |  |
| --- | --- | --- | --- |
| ***API*** | ***Description*** | ***Input arguments*** | ***Return Value*** |
| **LCD\_voidInit** | This function is used to initialize the LCD.  it sets the functionality of LCD to be 8 bit mode or 4 bit mode .  It also sets the LCD initial state of the LCD. | None | None |
| **LCD\_voidWriteCommand** | This function used to send a command to the LCD internal controller. | A variable that holds the 8 bit command to be sent | None |
| **LCD\_voidWriteChar** | This function used to send data to be displayed on LCD.  These data usually is the ASCII code of the character to be displayed, as the ASCII code is considered the physical address of this character inside the CGROM of the LCD controller | A variable that holds the 8 bit data to be displayed | None |
| **LCD\_voidWriteString** | This function used to send string to be displayed on LCD by sending displaying character by character.  So this function calls the **LCD\_voidWriteChar()** function as many times as the number of characters inside the string ( character array). | a pointer to string (char array). | None |
| **LCD\_voidWriteCustomChar** | this function is used to write a custom character to the CGRAM of the LCD to be displayed later on. | A pointer to array that holds the character, each element consider the value of a Row in a custom character and a 8-bit variable that holds the location of the created character in CGRAM | None |
| **LCD\_voidClearLine** | This function is used to clear a specific line in LCD. | A 8-bit variable that holds the number of line to be cleared the first or second line. | None |

Table 2: API description

**Dependant Modules API’s:**

* DIO driver:

|  |  |  |  |
| --- | --- | --- | --- |
| **DIO\_u8WritePinVal** | API which is used to write the values of bits of the characters or command to the DIO pins that connected to the LCD pins to write them on the LCD screen. | DIO pin index and the value of pin high or low. | The API returns a status of the operation whether it is ok or not. |

|  |  |  |
| --- | --- | --- |
| **Variable** | **Type** | **Description** |
| **LCD\_u8dataPins[ ]** | Global u8 | This array holds the data pins of the LCD that refers to a digital output pins on the microcontroller |
| **Local\_u8Counter** | Local u8 | It is a loop counter that loops over the 8 pins of the LCD to write the values of command or character values on the DIO pins corresponding to the LCD pins. |
| **local\_u8Char\_address** | Local u8 | This variable is used for accessing the CGRAM location it holds the address of the required CGRAM location to write certain character to CGRAM. |

Table 3: Variables of the component

Design Constrains

## Constraints on Initialization

* The configuration of data and control pins must be right to avoid wrong initialization and the modes of the LCD, which affect the initialization process.
* This module needs the DIO driver to be initialized first, because the DIO driver API’s are used in side this module.
* Make sure to include the shared libraries used
* Types.h.

## Constraints on Inputs

While Writing Custom character to CGRAM make sure that the 8 bit variable that holds the location of the created custom character in CGRAM to be in range from 0 to 7.

# Configuration

|  |  |  |
| --- | --- | --- |
| **Configuration Parameters** | **Description** | **Comment** |
| Data pins | DIO pins connected to LCD data pins. | 8 pins for data in range from pin0 to pin31. |
| Control pins | DIO pins connected to LCD control pins. | 8 pins for data in range from pin0 to pin31. |
| Initial state | The initial state of the LCD. | To be ON or OFF and show the cursor of not. |
| LCD Mode | The LCD data mode. | To be 4-bit or 8-bit mode. |

Table 4: Configuration Parameters

## File Description

|  |  |
| --- | --- |
| **File Name** | **Description** |
| LCD\_interface.h | It contains the prototypes of the APIs. |
| LCD\_config.h | It contains the configuration parameters of the component. |
| LCD\_private.h | It contains private configuration parameters. |
| LCD\_prog.c | It contains the written code of the APIs. |

Table 5: Files Description

{END\_VAL\_SPECIFICATION\_DOC}