The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

AN205514

F²MC-8FX Family MB95200 Series 8-Bit Microcontroller LED API

Associated Part Family: MB95200 Series

This document introduces API for LED driven by MB95200.

1 Introduction

This document introduces API for LED driven by MB95200.

I/O port is used to drive LED and send different values to display different numbers.

2 Function List in LED Library

This chapter introduces the functions in LED library.

Table 1 lists function in the LED library.

Table 1. LED Function

Function name	Description
void Led_Funct(unsigned char Show_Num)	Show the number rang is 0~9

3 Details of LED Function

This chapter introduces the details of LED function.

3.1 Led_Funct Function

Table 2 describes the Led_Funct function.

Table 2. Led_Funct Function

Function name	Led_Funct
Function prototype	void Led_Funct(unsigned char Show_Num)
Function description	Display number from 0~9 on LED, use I/O port0
Input parameter1	Show_Num, set the number which need to display
Return value	None
Example	Display number 5: Led_Funct(0x05);

This function is used to display the number which needed to show.

Table 3 describes parameter values of Show_Num.

Table 3. Show_Num Definition

Show_Num	Description
0x00	Display number 0
0x01	Display number 1
0x02	Display number 2
0x03	Display number 3
0x04	Display number 4
0x05	Display number 5
0x06	Display number 6
0x07	Display number 7
0x08	Display number 8
0x09	Display number 9

3.2 Array

There is one array in function Led_Funct(). It lists all values which will drive port0.

Table 4 describes the array Data_Lib[10].

Table 4. Array Definition

Array name	Data_Lib[10]
Array prototype	unsigned char Data_Lib[10]
Array description	List all value which will be use in drive function
Value	Hex value which drive the port to let the led show interrelated number
Example	For general positive polarity LED: unsigned char Data_Lib[10]= {0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};

For general positive polarity LED, setting the pin to low will light the led; for general negative polarity LED, setting the pin to high will light the led.

Table 5 describes hex values in general positive polarity LED.

Table 5.Positive Polarity Definition

Array Parameter	Description General Positive LED
0xc0	Display number 0
0xf9	Display number 1
0xa4	Display number 2
0xb0	Display number 3
0x99	Display number 4
0x92	Display number 5
0x82	Display number 6
0xf8	Display number 7
0x80	Display number 8
0x90	Display number 9

Table 6 describes hex values in general negative polarity LED.

Table 6. Negative Polarity Definition

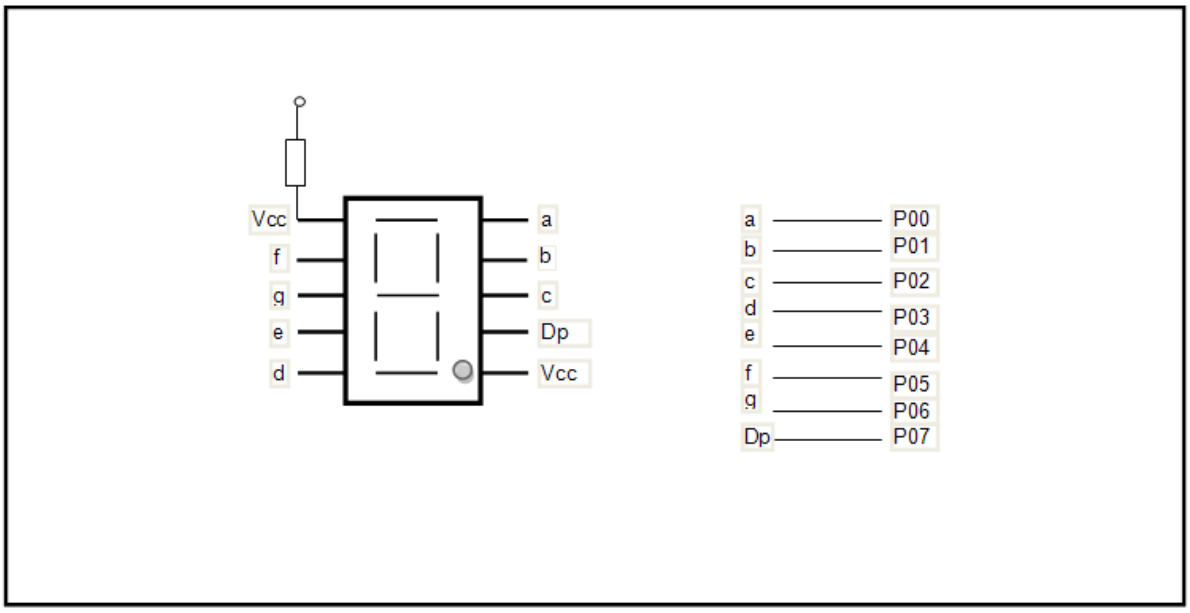
Array Parameter	Description General Negative LED
0x3f	Display number 0
0x06	Display number 1
0x5b	Display number 2
0x4f	Display number 3
0x66	Display number 4
0x6d	Display number 5
0x7d	Display number 6
0x07	Display number 7
0x7f	Display number 8
0x6f	Display number 9

4 Hardware

This chapter describes the hardware.

Figure 1 describes the instance port0 driving one positive polarity LED.

Figure 1. Hardware Definition



5 Additional Information

For more information on Cypress MB95200 products, please visit following website:

http://www.cypress.com/MB95200

Document History

Document Title: AN205514 – F²MC-8FX Family MB95200 Series 8-Bit Microcontroller LED API

Document Number: 002-05514

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	HUAL	04/08/2009	Initial release
*A	5265901	HUAL	05/18/2016	Migrated Spansion Application Note MCU-AN- 500040-E-10 to Cypress format.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corporation. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2009-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.
TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.
Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.