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.

AN205510

F²MC-8FX 家族 MB95200 系列使用矩阵的键盘开发

使用 MB95200 系列 MCU 设计键盘功能的方法一共有三种：外部中断，AD 以及矩阵。本文档描述了如何使用矩阵法设计键盘功能，并以实例阐述了该方法。

目录

1 概要 1
2 硬件设计 1
3 键盘开发 2
3.1 矩阵电路的功能 2
3.2 抖动消除 2
3.3 抖动消除示例 2
4 资源使用 3
4.1 寄存器介绍 3
4.2 简介 4
5 软件设计 5
5.1 软件设计 5
6 代码示例 7
7 性能评估 10
8 附加信息 11
文档修改记录 12

1 概要

使用 MB95200 系列 MCU 设计键盘功能的方法一共有三种：外部中断，AD 以及矩阵。

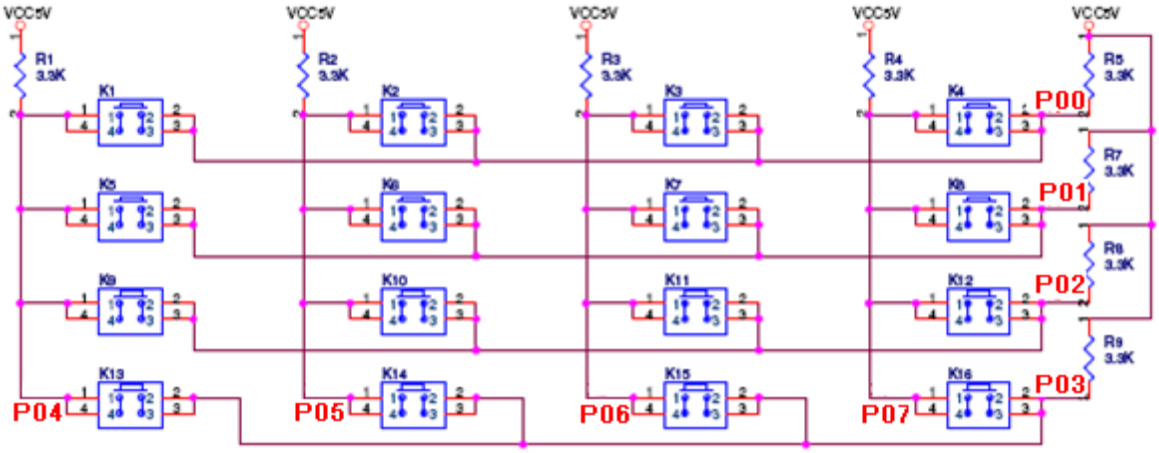
本文档描述了如何使用矩阵法设计键盘功能，并以实例阐述了该方法。

2 硬件设计

本章介绍了如何运用矩阵法使用硬件电路设计键盘功能。

目前有两种方法可用于消除抖动：软件法和硬件法。要使用硬件法，必须在电路中增加一个电容。

图 1. 使用矩阵设计键盘的电路



如上图所示，P04~P07 用作输出引脚，P00~P03 用作输入引脚，用于扫描键盘状态。

3 键盘开发

本章介绍了如何使用矩阵开发键盘。

3.1 矩阵电路的功能

键盘是输入系统的一个重要工具。本参考设计指出在采用具有许多按键的键盘时如何精简 I/O 引脚。

本设计采用了检测键盘状态的实时监控法。检测周期是一个固定值，这意味着任何比该值慢的动作都将被监控软件捕捉，从而确保了键盘的高响应性。

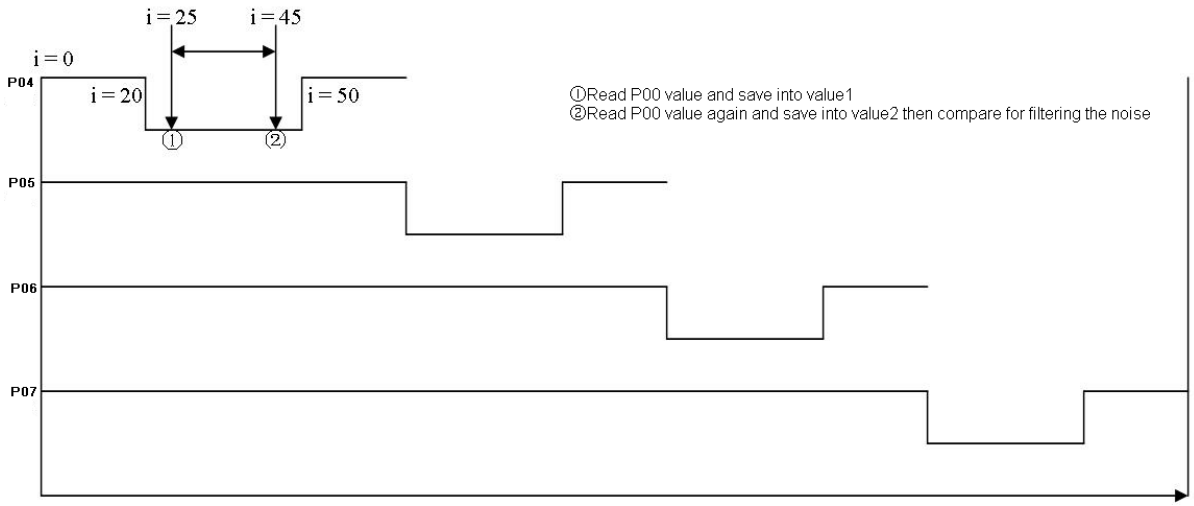
3.2 抖动消除

抖动是键盘设计中存在的一个问题，有二种方法可以解决这个问题。一种是在硬件电路中增加一个电容，另一种是通过使用 8/16 位多功能定时器增加延时。

3.3 抖动消除示例

以下时序图是消除抖动的一个示例。该示例采用矩阵法设计一个 4x4 的键盘。X0、X1、X2 和 X3 依次扫描键盘 50ms，扫描整个键盘需要大约 200ms。

图 2. 抖动消除的时序图



4 资源使用

本章介绍了正常运行状态下的评估步骤。

要监控键盘状态，必须使用 I/O 引脚。本章将介绍 I/O 端口的使用。参见 MB95200 系列硬件手册的第 9 章了解关于寄存器设置的更多信息

4.1 寄存器介绍

4.1.1 端口数据寄存器（PDR）

该寄存器包含对应输入或输出引脚的位信息。如果端口方向寄存器设置为输出模式，则输出该值。

注意：资源输出控制位重写 PDR 位值。

PDRx_yz	Pin 功能
0	Pin state low (VSS)
1	Pin state high (VDD)

4.1.2 数据方向寄存器（DDR）

如果引脚用作输入或输出，该寄存器包含对应引脚的位信息。

DDRx_yz	外围功能输出	引脚功能
0	禁用	端口输入
1	禁用	端口输出
invalid	启用	外围功能输出

4.1.3 上拉控制寄存器（PUL）

该寄存器连接一个内部上拉电阻至端口引脚。

PULx_yz	上拉电阻
0	禁用
1	启用

参见数据表了解关于电阻值的更多信息。

4.1.4 输入电平选择寄存器（ILSR）

该寄存器可以选择以下输入电平。

ILSR	输入电平	VIL	VIH
0x04	CMOS	0.3 VCC	0.7 VCC
0x00	迟滞	0.3 VCC	0.7 VCC

请注意只有 PDR0_P04 有此功能。

4.2 简介

本系统的 I/O 引脚可以分为两类：输出引脚和输入引脚。输出引脚用于输出高电压，输入引脚用于扫描键盘状态。

4.2.1 如何设置输入引脚

如果引脚用作输入端口，数据方向寄存器的对应位应设置为“0”。

要设置外部连接源至高 Z 状态，请使用外部上拉或下拉电阻器或者设置上拉寄存器的对应位。

输入模式有三种类型：数字输入、ADC 输入以及外围功能输入。

- 数字输入表示该端口用作通用 I/O。
- ADC 输入表示该端口仅用作模拟输入。
- 外围功能输入表示该端口被外围功能用作输入，例如外部中断输入。

4.2.2 如何设置输出引脚

输出模式有两种类型：数字输出和外围功能输出。

- 数字输出表示该端口用作通用 I/O。
- 外围功能输出表示该端口用作外围资源输出，例如 8/16 位多功能定时器的输出。

4.2.3 上拉寄存器

在输入模式下，P0 和 PG 端口可通过编程上拉寄存器（2.2.3）启用内部上拉电阻（大约 50KΩ，参见数据表了解其具体取值）。

初始值“0”断开与内部上拉电阻的连接。 烧写“1”至 PULx 寄存器的对应位启用电阻。

如果端口用作输出端口，寄存器位的值没有意义，上拉电阻被禁用。（例外：对于 I²C 引脚 SDA 和 SCL，设置保留。对于 UART 输出 SOT，如果不是由线路驱动器提供，则可以使用内部上拉电阻。）

微型控制器处于停止或定时器模式下时，上拉电阻被禁用。

如果引脚用作 ADC 输入，电阻也被禁用。

如果外部引脚被外部总线接口使用，则不能使用内部上拉寄存器。

5 软件设计

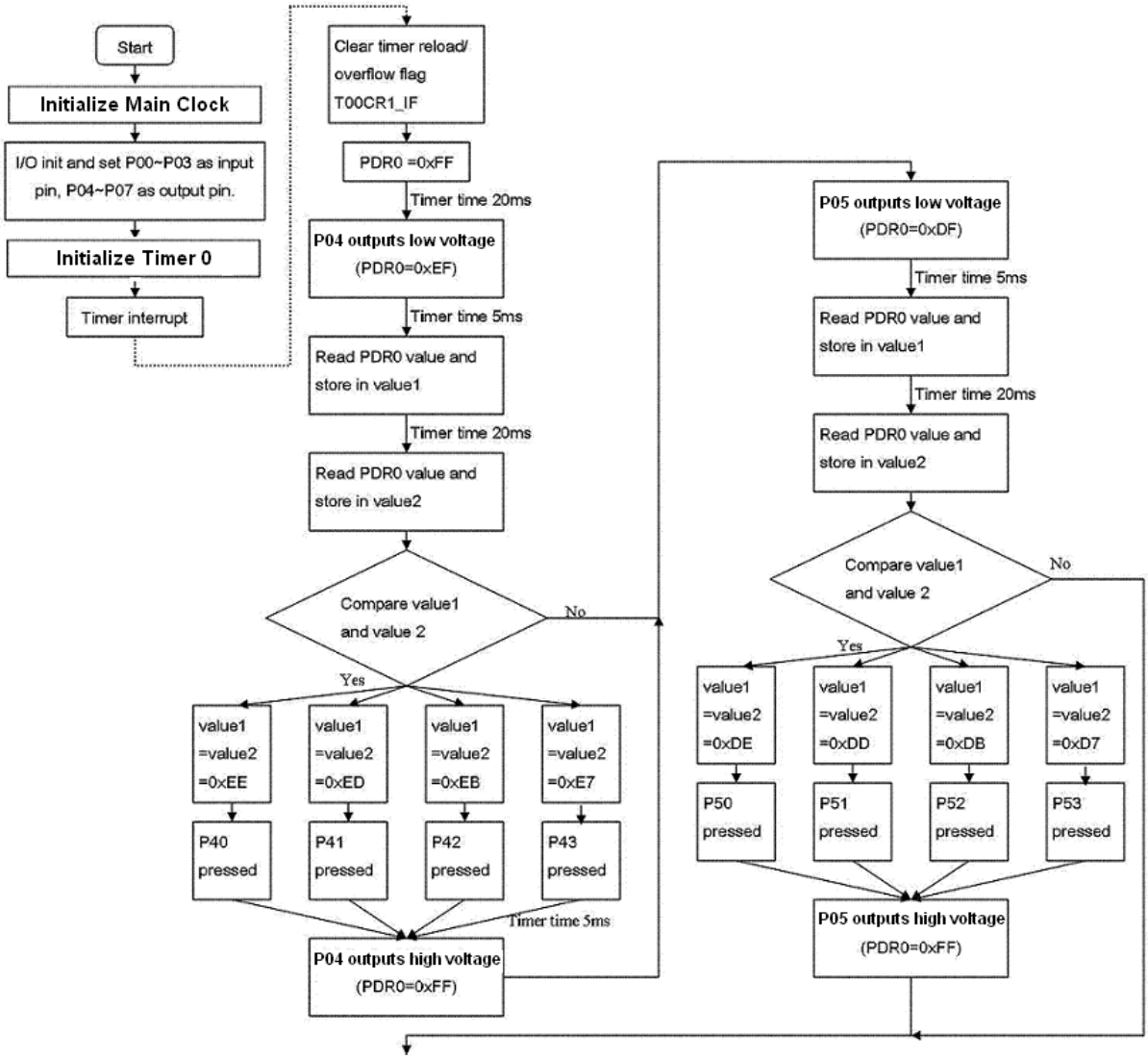
本章描述了如何使用矩阵设计键盘。

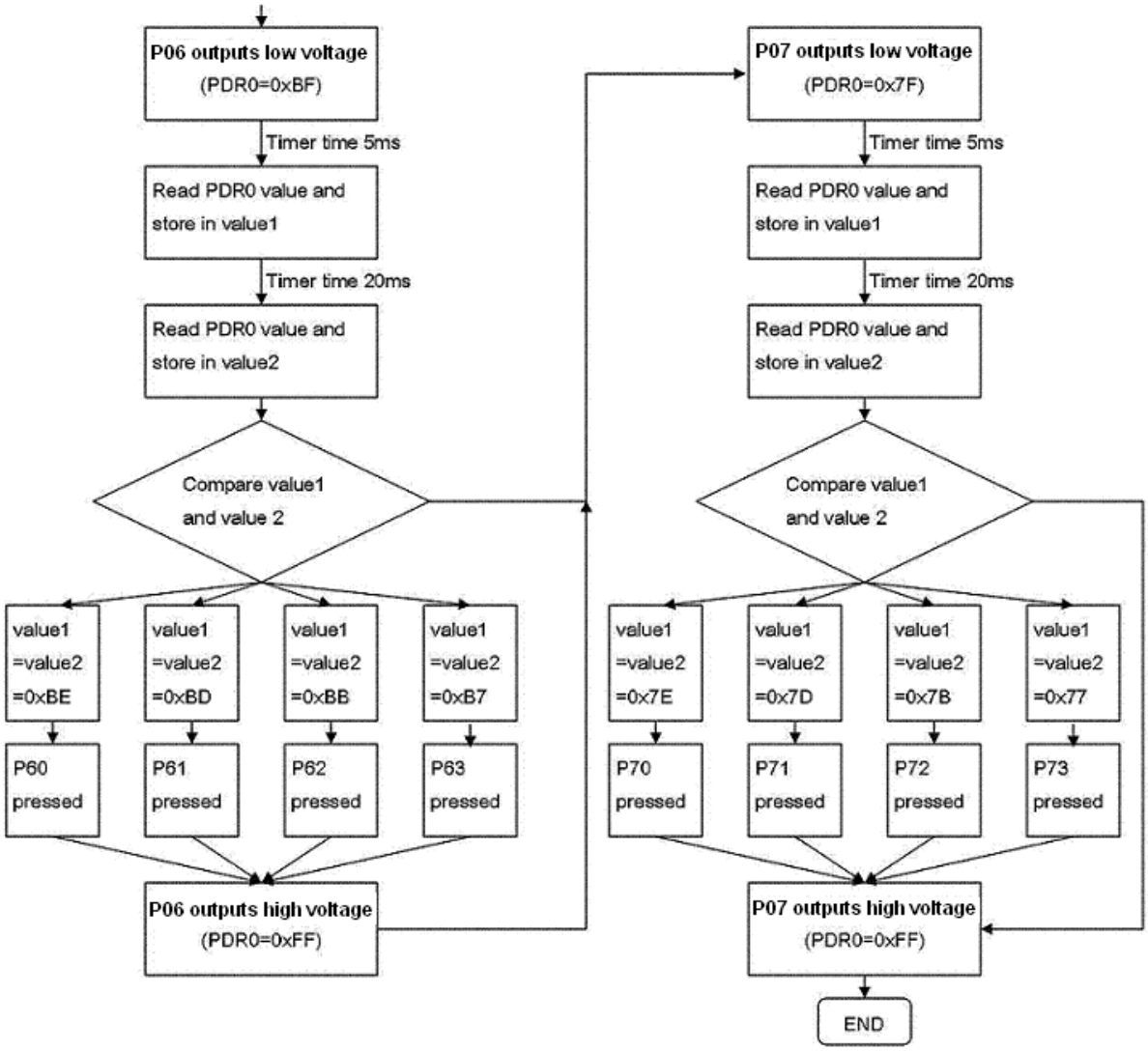
5.1 软件设计

要实现此功能，首先初始化 I/O 寄存器，并禁用 AD 输入，然后启用定时器中断消除抖动。输出 I/O 引脚输出高电压，MCU 等待输入信号。参见硬件电路，如果按键被按下，MCU 将扫描该信息，并执行对应的按键功能。

流程图如下所示：

图 3. 使用矩阵设计键盘的流程图




P06 outputs low voltage
(PDR0=0xBF)
Timer time 5ms
Read PDR0 value and store in value1
Timer time 20ms
Read PDR0 value and store in value2
Compare value1 and value 2
value1 =value2 =0xBE
value1 =value2 =0xBD
value1 =value2 =0xBB
value1 =value2 =0xB7
P60 pressed
P61 pressed
P62 pressed
P63 pressed
P06 outputs high voltage
(PDR0=0xFF)
P07 outputs low voltage
(PDR0=0x7F)
Timer time 5ms
Read PDR0 value and store in value1
Timer time 20ms
Read PDR0 value and store in value2
Compare value1 and value 2
value1 =value2 =0x7E
value1 =value2 =0x7D
value1 =value2 =0x7B
value1 =value2 =0x77
P70 pressed
P71 pressed
P72 pressed
P73 pressed
P07 outputs high voltage
(PDR0=0xFF)
END

6 代码示例

本章描述了使用矩阵开发键盘的代码示例。

以下代码基于 MB2146-410-01，说明了如何使用 I/O 端口开发键盘功能。端口 0 用于检测按键状态。输入默认级别为高。

```
/* Give a example for basic I/O matrix */
#include "../MB95200_IO/mb95200.h"
void timer_init(void)
{
T01DR = 0x13;       // 5000us
T00DR = 0x88;
TMCR0 = 0x10;       // 16-bit
T00CR0 = 0x81;      // interval timer with continuous mode
T00CR1 = 0xA0;      // disable output, start timer
}
/*P04 SCAN*/
void IO_restart(void)      // PDR0 restart
{
PDR0 = 0xFF;
}
void P04_low(void)                // Pin P04 begin to scan
{
PDR0_P04 = 0;
}
void Port_Value1(void)     // Read and store PDR0 in value 1
{
port_value1 = PDR0;
}
void Row_one_process(void) // key judge
{
port_value2 = PDR0; // Read and store PDR0 in value 2
if(port_value2 == port_value1)
{
switch(port_value2)
{
case 0xEE: SW1();
break;
case 0xED: SW2();
break;
case 0xEB: SW3();
break;
case 0xE7: SW4();
break;
}
}
}
/* P05 scan*/
void P05_low(void)                // Pin P05 begin to scan
{
PDR0_P05=0;
}
```

```
void Row_two_process(void)        // key judge
{
port_value2 = PDR0;
if(port_value2 == port_value1)
{
switch(port_value2)
{
case 0xDE: SW5();
break;
case 0xDD: SW6();
break;
case 0xDB: SW7();
break;
case 0xD7: SW8();
break;
}
}
}
/* P06 scan*/
void P06_low(void)                        // Pin P06 begin to scan
{
PDR0_P06=0;
}
void Row_three_process(void)      // key judge
{
port_value2 = PDR0;
if(port_value2 == port_value1)
{
switch(port_value2)
{
case 0xBE: SW9();
break;
case 0xBD: SW10();
break;
case 0xBB: SW11();
break;
case 0xB7: SW12();
break;
}
}
}
/* P07 scan*/
void P07_low(void)                        // Pin P07 begin to scan
{
PDR0_P07=0;
}
void Row_four_process(void)       // key judge
{
port_value2 = PDR0;
if(port_value2 ==port_value1)
{
switch(port_value2)
{
case 0x7E: SW13();
break;
case 0x7D: SW14();
break;
case 0x7B: SW15();
break;
case 0x77: SW16();
break;
}
}
```

```
void SW1(void)                  // Key Process
{
. . .
}
. . .
. . .
void SW16(void)
{
. . .
}
. . .
__interrupt void Timer_Interrupt (void) //key scan process
{
T00CR1_IF=0;
timer_counter++;
if(timer_counter==0)  // PDR0 restart
{
IO_restart();
}
if(timer_counter==4)  // P04 begin to scan
{
P04_low();
}
if(timer_counter==5)  // Read and store PDR0
{
Port_Value1();
}
if(timer_counter ==9)          // Read and store PDR0 and judge keypressed
{
Row_one_process();
}
if(timer_counter==10)          // PDR0 restart
{
IO_restart();
}
if(timer_counter==14)          // P05 begin to scan
{
P05_low();
}
if(timer_counter==15)          // Read and store PDR0
{
Port_Value1();
}
if(timer_counter==19)          // Read and store PDR0 and judge keypressed
{
Row_two_process();
}
if(timer_counter==20)          // PDR0 restart
{
IO_restart();
}
if(timer_counter==24)          // P06 begin to scan
{
P06_low();
}
if(timer_counter==25)          // Read and store PDR0
{
Port_Value1();
}
```

```
if(timer_counter==29) // Read and store PDR0 and judge keypressed
{
Row_three_process();
}
if(timer_counter==30) // PDR0 restart
{
IO_restart();
}
if(timer_counter==34) // P07 begin to scan
{
P07_low();
}
if(timer_counter==35) // Read and store PDR0
{
Port_Value1();
}
if(timer_counter==39) // Read and store PDR0 and judge keypressed
{
Row_four_process();
}
if(timer_counter==40) // PDR0 restart
{
IO_restart();
}
if(timer_counter>=41) // Timer_counter restart
{
timer_counter=0;
}
}
void clock_Select(void)
{
SYCC = 0x00;
WATR = 0x00;
STBC = 0x01;
SYCC2= 0xF4;
}
void main(void)
{
InitIrqLevels();
__EI();
timer_counter =0;
DDR0 = 0xF0;                //P00~P03= in P04~P07=out
AIDRL = 0xFF;       //Port input enable
clock_Select();     //Main clock select
timer_init();       //Timer initialize
while(1);
}
```

7 性能评估

如上所述，消除抖动的方法有两种：硬件法和软件法。尽管硬件法的代码非常简单，仅使用定时器计时器的效果并不好，因为系统不能扫描沿。比较起来，软件方法更加有用。

8 附加信息

如欲了解有关如何使用 MB9595200H / 210H EV-板、BGM 适配器和 SOFTUNE 的更多信息，敬请参见 SKT MB2146-410A -01 -E 用户手册，或者访问以下网址：

http://www.cypress.com/documentation/software-and-drivers/f2mc-8fx-mb95200h210h-series-starter-kit-mb2146-410a-01-e-setup

文档修改记录

文档标题：AN205510 - F²MC-8FX 家族 MB95200 系列使用矩阵的键盘开发

文档编号：002-05776

修订版	ECN	变更者	提交日期	变更说明
**	—	HUAL	05/03/2009	初稿
*A	5350920	HUAL	07/14/2016	已将 Spansion 应用手册《MCU-AN-500038-Z-10》转换成 Cypress 格式。

全球销售和设计支持

赛普拉斯公司拥有一个由办事处、解决方案中心、厂商代表和经销商组成的全球性网络。如果想要查找离您最近的办事处，请访问赛普拉斯所在地。

产品

ARM® Cortex® 微控制器	cypress.com/arm
汽车级	cypress.com/automotive
时钟与缓冲器	cypress.com/clocks
接口	cypress.com/interface
照明和电源控制	cypress.com/powerpsoc
存储器	cypress.com/memory
PSoC	cypress.com/psoc
触摸感应	cypress.com/touch
USB 控制器	cypress.com/usb
无线/射频	cypress.com/wireless

PSoC® 解决方案

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

赛普拉斯开发者社区

论坛 | 项目 | 视频 | 博客 | 培训 | 组件

技术支持

cypress.com/support

PSoC 是赛普拉斯半导体公司的注册商标。PSoC Creator 是赛普拉斯半导体公司的商标。 此处引用的所有其他商标或注册商标都归其各自所有者所有。

赛普拉斯半导体
198 Champion Court
San Jose, CA 95134-1709

电话 : 408-943-2600
传真 : 408-943-4730
网站地址 : www.cypress.com

©赛普拉斯半导体公司，2009-2016 年。本文件是赛普拉斯半导体公司及其子公司，包括 Spansion LLC（“赛普拉斯”）的财产。本文件，包括其包含或引用的任何软件或固件（“软件”），根据全球范围内的知识产权法律以及美国与其他国家签署条约由赛普拉斯所有。除非在本款中另有明确规定，赛普拉斯保留在该等法律和条约下的所有权利，且未就其专利、版权、商标或其他知识产权授予任何许可。如果软件并不附随有一份许可协议且贵方未以其他方式与赛普拉斯签署关于使用软件的书面协议，赛普拉斯特此授予贵方属个人性质的、非独家且不可转让的如下许可（无再许可权）（1）在赛普拉斯特软件著作权项下的下列许可权（一）对以源代码形式提供的软件，仅出于在赛普拉斯硬件产品上使用之目的且仅在贵方集团内部修改和复制软件，和（二）仅限于在有关赛普拉斯硬件产品上使用之目的将软件以二进制代码的形式向外部最终用户提供（无论直接提供或通过经销商和分销商间接提供），和（2）在被软件（由赛普拉斯公司提供，且未经修改）侵犯的赛普拉斯专利的权利主张项下，仅出于在赛普拉斯硬件产品上使用之目的制造、使用、提供和进口软件的许可。禁止对软件的任何其他使用、复制、修改、翻译或汇编。
在适用法律允许的限度内，赛普拉斯未对本文件或任何软件作出任何明示或暗示的担保，包括但不限于关于适销性和特定用途的默示保证。在适用法律允许的限度内，赛普拉斯保留更改本文件的权利，届时将不另行通知。赛普拉斯不对因应用或使用本文件所述任何产品或电路引起的任何后果负责。本文件，包括任何样本设计信息或程序代码信息，仅为参考之目的提供。文件使用人应负责正确设计、计划和测试信息应用和由此生产的任何产品的功能和安全性。赛普拉斯产品不应被设计为、设定为或授权用作武器操作、武器系统、核设施、生命支持设备或系统、其他医疗设备或系统（包括急救设备和手术植入物）、污染控制或有害物质管理系统中的关键部件，或产品植入之设备或系统故障可能导致人身伤害、死亡或财产损失的其他用途（“非预期用途”）。关键部件指，若该部件发生故障，经合理预期会导致设备或系统故障或会影响设备或系统安全性和有效性的部件。针对由赛普拉斯产品非预期用途产生或相关的任何索赔、费用、损失和其他责任，赛普拉斯不承担全部或部分责任且贵方不应追究赛普拉斯之责任。贵方应赔偿赛普拉斯因赛普拉斯产品任何非预期用途产生或相关的所有索赔、费用、损失和其他责任，包括因人身伤害或死亡引起的索赔，并使之免受损失。
赛普拉斯、赛普拉斯徽标、Spansion、Spansion 徽标，及上述项目的组合，及 PSoC、CapSense、EZ-USB、F-RAM 和 Traveo 应视为赛普拉斯在美国和其他国家的商标或注册商标。敬请访问 cypress.com 获取赛普拉斯商标的完整列表。其他名称和品牌可能由其各自所有者主张为该方财产。