



**成都国腾微电子有限公司**

**GM8166**

**32 位并行 I/O 口扩展芯片**

**应用手册 2003.11**

GM8166		
版本记录：1.0		当前版本时间：2003 年 11 月
新旧版本改动比较：		
旧版 文档页数	当前版本 文档页数	主题（和旧版本相比的主要变化）

如果您有技术、交付或价格方面的任何问题，请联系成都国腾微电子有限公司的相关办公室或当地的代理商，或访问我们的网站：[www.gticc.com](http://www.gticc.com)，谢谢！

编制时间：2003 年 11 月  
由成都国腾微电子有限公司发布  
发布地点：成都  
成都国腾微电子有限公司版权所有。

## 1 概述

GM8166 芯片通过串入并出、并入串出、并入并出转换完成 I/O 口的扩展。最高工作频率为 10MHz，用于配合 MCU 完成对多个外围电路的控制和信号采集工作。同时具备 SPI 总线接口，满足了更多需求。设计时充分考虑了抗电磁干扰和工作温度范围，完全适用于工业领域。

## 2 应用领域

通信设备：IC 卡话机，雷达控制

安防电路：报警器，消防控制

仪器仪表：电表控制，多路采集

工业控制：印刷机械控制，注塑机控制，机车控制，相机控制

金融机具：POS 机，IC 卡机具

## 3 应用示例

### 3.1 串并/并串转换应用示例

此电路利用单片机 89C51 串口工作方式 0 来进行单片机的 I/O 口电路的扩展。

外部接口应用如图 1 所示：

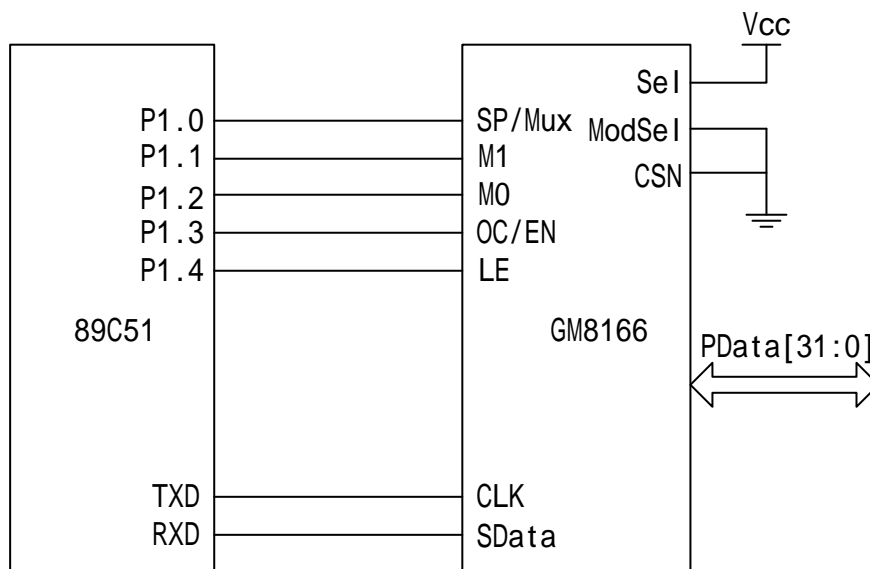


图 1 典型应用电路

说明：GM8166 的各个控制引脚由 89C51 控制，可在 8166 的各工作模式间进行转换。若只需使用一种或几种工作模式，可将不用的控制引脚置入所需电平，无需接入 MCU，例如此例中因为只用 1 片 8166，所以将 CSN 固定接地，同时 8166 只工作在串并/并串转换模式下，所以将 ModSel 引脚固定接地。

软件示例：

```
/*
*****
*/
#include <reg51.h>
/*
*****I/O 定义*****
*/
sbit S_P=P1^0;           //串并/并串模式选择信号
sbit M1=P1^1;           //工作方式控制信号 1
sbit M0=P1^2;           //工作方式控制信号 0
sbit OC =P1^3;          //并行输出/输入允许信号
sbit LE=P1^4;           //并行输出数据锁存信号
sbit CLK = P3^1;
/*
*****数据变量定义*****
*/
unsigned char SendBuff[4]={0xff,0xff,0x67,0xbc };
unsigned char ReceiveBuff [2]={0x00,0x00};
unsigned char i=0 ,j=0 ;
/*
*****主程序*****
*/
void main(void)
{
SCON=0x00;               //设置 MCU 串行口工作方式为方式 0
PCON=0x00;

/* 32 位串入并出转换 */
S_P=1;
M1=1;
M0=1;                   //设定 GM8166 工作模式为 32 位的串入并出模式
OC=1;                   //输出禁止
LE=1;                   //输出锁存开

for (i=0;i<4;i++)
{
SBUF= SendBuff[i];      //发送数据到 GM8166
while(TI==0);
TI=0;
}

LE=0;                   //输出数据锁存
OC=0;                   //输出允许

/* 16 位并入串出转换 */
S_P=0;
M1=1;
M0=0;                   //设定 GM8166 工作模式为 16 位的并入串出模式
OC=0;                   //并行输入允许

CLK=0;
CLK=1;                   //并行口数据置入寄存器

```

```

OC=1;                //并行输入禁止

for (j=0;j<2;j++)
{
    REN=1;
    while(RI==0);
    ReceiveBuff [j]=SBUF;    //串行数据输入 89C51
    RI=0;
    REN=0;
}
}
    
```

### 3.2 级连功能应用示例

数据传输超过 32 位时，用户可通过 CData 口实现级连输入/输出，达到位数的扩展。  
 例如用户需要进行 40 位数据的转换，用两片 8166 实现。  
 硬件接口见图 2 所示：

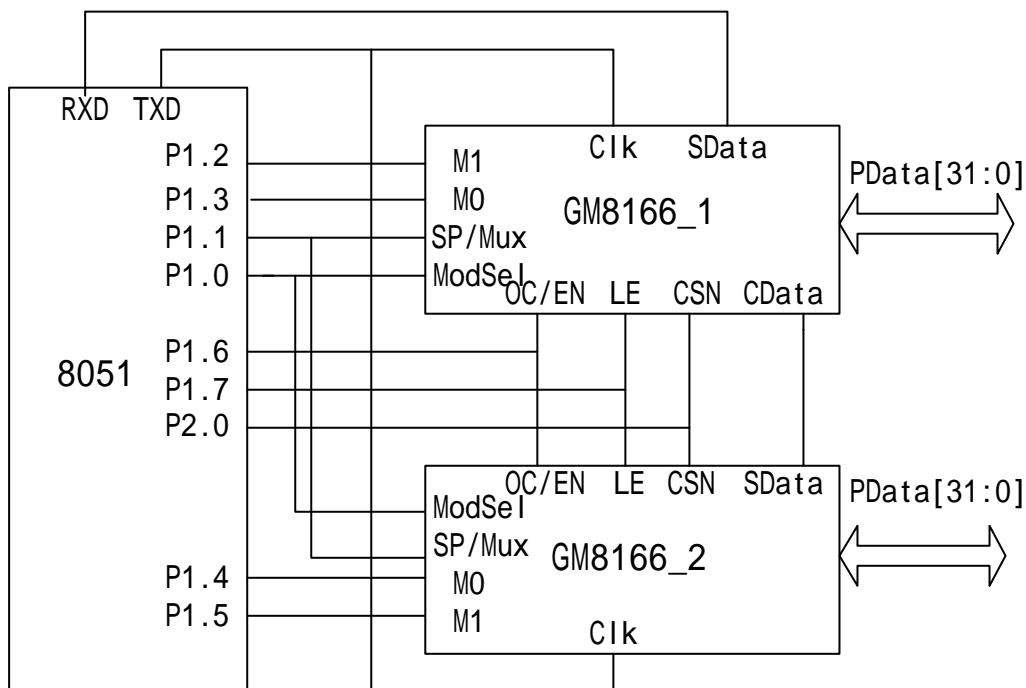


图 2 级连应用电路

说明：若只需使用 GM8166 的一种或几种工作模式，可将不用的控制引脚置入所需电平，无需接入 MCU。例如上图中 CSN、ModSel 引脚可以根据需要直接接地。

软件示例（以串入并出为例）：

```

/*****/
#include <reg51.h>
/*****I/O 定义*****/
sbit ModSel=P1^0;    //并串/串并转换模式和并口切换模式选择信号
    
```

```

sbit S_P=P1^1;           //串并/并串模式选择信号
sbit M1_1=P1^2;         //GM8166_1 的工作方式控制信号 1
sbit M0_1=P1^3;         // GM8166_1 的工作方式控制信号 0
sbit M0_2=P1^4;         // GM8166_2 的工作方式控制信号 0
sbit M1_2=P1^5;         // GM8166_2 的工作方式控制信号 1
sbit OC =P1^6;          //并行输出/输入允许信号
sbit LE=P1^7;           //并行输出数据锁存信号
sbit CSN = P2^0;        //片选信号
/*****数据变量定义*****/
unsigned char SendBuff[5]={0xaa,0xc9,0x67,0xbc,0x55 };
unsigned char j=0 ;
/*****主程序*****/
void main(void)
{
SCON=0x00;              // 设置 MCU 串行口工作方式为方式 0
PCON=0x00;

ModSel=0;
S_P=1;
M1_1=1;
M0_1=1;                 //设定 8166-1 工作模式为 32 位的串入并出模式
M0_2=0;
M1_2=0;                 //设定 8166-2 工作模式为 8 位的串入并出模式
OC =1;                  //输出禁止
LE=1;                   //输出锁存开
CSN=0;                  //片选有效

for (j=0;j<5;j++)
{
SBUF= SendBuff [j];    //发送数据到 8166
while(TI==0);
TI=0;
}

LE=0;                   //输出数据锁存
OC=0;                   //允许输出
}

```

### 3.3 并口切换输入/输出应用示例

GM8166 提供并口切换输入/输出功能，利用该项功能，可以实现 3 路 8 位口的开关切换。以 89C51 单片机为例，硬件接口示例图见图 3：

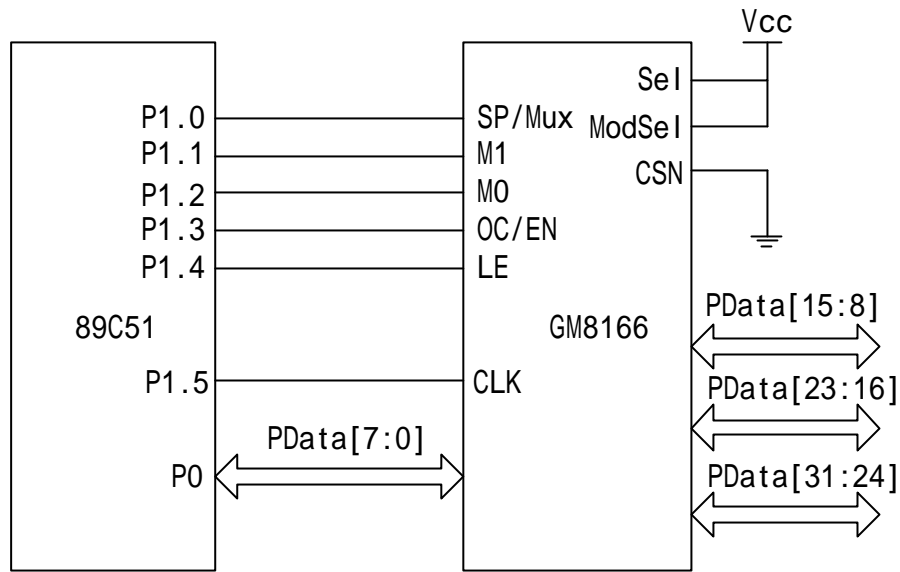


图 3 并口切换应用电路

说明：若 GM8166 只做单一用途，不需要改变的控制引脚可以不用 MCU 控制，直接置入其所需电平即可。

软件示例：

```

/*****
#include <reg51.h>
/*****I/O 定义*****/
sbit Mux =P1^0;          //1 路输入 3 路输出和 3 路输入 1 路输出选择信号
sbit M1=P1^1;           //GM8166 的工作方式控制信号 1
sbit M0=P1^2;           // GM8166 的工作方式控制信号 0
sbit OC=P1^3;           //并行输出允许信号
sbit LE=P1^4;           //并行输出数据锁存信号
sbit CLK = P1^5;        //时钟信号
/*****主程序*****/
void main(void)
{
/*1 路输入、3 路选择输出 */
Mux=0;                  //选择 1 路输入，3 路输出模式
M1=1;
M0=0;                   //选择 I/O[15：8]口输出
OC=1;                   //禁止输出
LE=1;                   //输出锁存开
P0=0xaa;                //P0 口输出数据

CLK=0;
CLK=1;                  //数据置入寄存器

LE=0;                   //并行输出数据锁存
OC=0;                   //并行输出允许开

```

```
M1=0;
M0=1;           //选择 I/O[23 : 16]口输出

P0=0x55;       //P0 口输出数据

CLK=0;
CLK=1;         //数据置入寄存器

LE=1;          //并行输出数据进入锁存器
LE=0;          //并行输出数据锁存

/* 3 路选择输入、1 路输出 */
Mux=1;         //选择 3 路输入，1 路输出模式
M1=1;
M0=1;         //选择 I/O[31 : 24]口输入
OC=1;         //禁止输出
LE=1;         //输出锁存开

CLK=0;
CLK=1;         //数据置入寄存器

LE=0;         //并行输出数据锁存
OC=0;         //并行输出允许开
}
```