

文章编号:1001-9081(2006)11-2759-03

实时操作系统 $\mu\text{C}/\text{OS-II}$ 在 DSP TMS320LF2407A 上的移植、裁剪及应用

谢 蓉,赵凯瑞,王新民

(西北工业大学 自动化学院,陕西 西安 710072)

(xierong2005@tom.com)

摘要:介绍了 V2.76 版 $\mu\text{C}/\text{OS-II}$ 的新功能和工作原理以及 DSP 芯片 TMS320LF2407A 的特点,给出了将 $\mu\text{C}/\text{OS-II}$ 裁剪后移植到 TMS320LF2407A 的方法。在系统创建并运行多任务,测试了移植后的代码,实现了 CAN 通信、UART 通信、频率测量、模拟量测量、系统自检测等系统功能。

关键词:实时多任务操作系统;裁剪;移植;TMS320LF2407A;DSP

中图分类号: TP316.2 **文献标识码:** A

Application and development of TMS320LF2407A DSP based on RTOS $\mu\text{C}/\text{OS-II}$

XIE Rong, ZHAO Kai-rui, WANG Xin-min

(School of Automation, Northwestern Polytechnical University, Xi'an Shaanxi 710072, China;)

Abstract: The principle and new functions of $\mu\text{C}/\text{OS-II}$ were introduced. The characteristics of DSP TMS320LF2407A were analyzed. What's more, how to configure $\mu\text{C}/\text{OS-II}$ on TMS320LF2407A was given. In the end, several tasks, such as UART communication, CAN communication, CAP, AD and BIT, were realized in this system.

Key words: Real-Time Operating System(RTOS); configure; port; TMS320LF2407A; DSP

随着 DSP 等微处理芯片内部功能的大量增加和系统实时性、可靠性需求的提高,简单的前后台程序已经不能满足设计要求。代码占用空间小、执行效率高、配置简单灵活的实时多任务操作越来越多地应用到嵌入式系统设计中。 $\mu\text{C}/\text{OS-II}$ 是一个源码公开、免费的实时多任务操作系统。它易于学习,可以方便地移植到多种微处理器上。DSP 芯片 TMS320LF2407A 集实时信号处理和控制在一体,具有很好的特性。本文给出了将 $\mu\text{C}/\text{OS-II}$ 裁剪后移植到 TMS320LF2407A 的方法和过程。在初步移植成功的基础上,创建了 CAN 通信任务、UART 通信任务、模拟量采集任务、捕获任务、系统自检测等任务,测试操作系统的内核功能函数,同时完成了一个嵌入式系统应用产品。从结果看,该产品的各个功能稳定可靠,系统具有很好的实时性。

1 $\mu\text{C}/\text{OS-II}$ 特点及其工作原理

$\mu\text{C}/\text{OS-II}$ 是著名的、源码公开的实时操作系统,是专为嵌入式应用设计的,可用于各类 8 位、16 位及 32 位单片机或 DSP。它的源代码公开且可移植、可固化、可裁剪,其性能完全可以与商业产品竞争。笔者所用的版本是 2004 年发布的 V2.76 版本。 $\mu\text{C}/\text{OS-II}$ V2.76 比较老版本的 $\mu\text{C}/\text{OS-II}$, V2.52 增加了两个新的功能:一个是事件标志组 Event_Flag(),另一个是互斥量 Mutex()。Event_Flag 用于应用程序中会用到先对几个不同事件条件做“与”、“或”操作,然后再实现任务调度的情况,它使任务的调度从依赖于某个事件扩展到可以依赖于多个事件;Mutex() 在处理互斥条件相关的应用中更为方便,可用于解决多任务在处理共享资源时的优先级反转问题。

$\mu\text{C}/\text{OS-II}$ 是一个抢先式实时多任务操作系统,不支持时

间片轮转调度,其内核包括任务调度、时间管理、内存管理和资源管理四大部分,没有文件系统、网络接口和输入输出界面。在 $\mu\text{C}/\text{OS-II}$ 中,最多有 64 个任务优先级,系统占用 8 个,用户可创建 56 个任务。每个任务都是无限循环的,每个任务都处于以下五种状态之一:休眠态、就绪态、运行态、挂起态和中断态,如图 1 所示。



图 1 $\mu\text{C}/\text{OS-II}$ 中任务的状态

$\mu\text{C}/\text{OS-II}$ 的工作原理就是“近似地每时每刻总是让优先级最高的任务处于运行状态”。为了保证这一点,它在调用系统 API 函数、中断结束、定时中断结束时总是执行调度算法。原作者通过事先计算好数据,简化了运算量,通过精心设计就绪表结构,使得延时可预知。任务的切换是通过模拟一次中断实现的。

2 DSP 芯片 TMS320LF2407A 的特点

TMS320LF2407A DSP 芯片是在 C24X 芯片的基础上派生的,主要由三大部分组成,即 CPU、内存及外设。芯片采用了改进的哈佛结构,将数据总线与程序总线分离,使用了四级流水线作业。它的体系结构专门为实时信号处理而设计,将实时处理能力和控制器外设功能集于一身,具有很好的特性。

TMS320LF2407A DSP 芯片具有如下特点^[1]:

- 1) 处理能力强。指令周期最短为 25ns,运算能力达 40MIPS。
- 2) 片内具有较大的闪烁存储器。TMS320C20x 是最早使用闪烁存储器的 DSP 芯片。闪烁存储器具有比 ROM 灵活、

收稿日期:2006-05-21

作者简介:谢蓉(1982-),女,陕西西安人,硕士研究生,主要研究方向:计算机控制、飞行器控制;赵凯瑞(1977-),男(满族),黑龙江望奎人,博士研究生,主要研究方向:无人机自主飞行控制;王新民(1951-),男,江苏扬州人,教授,博士生导师,主要研究方向:计算机控制、飞行器控制研究、先进控制技术及仿真。

比 RAM 便宜的特点。利用闪烁存储器存储程序,不仅降低了成本,减小了体积,同时系统升级也较方便。

3) 功耗低。TMS320LF2407A DSP 芯片在 3.3V 工作时每个 MIPS 消耗 1.1mA 电流。使用 DSP 核的省电模式可进一步降低功耗。

4) 资源配置灵活。

TMS320LF2407A DSP 芯片具有丰富的 I/O 控制引脚和片上功能外设,如可配置数字 I/O 模块、16 通道的 10 位 A/D 转换器模块、串行通信接口 (SCI)、串行外设接口 (SPI)、事件管理器模块 (EV)、具有多达 16 个通道的 PWM (Pulse Width Modulation)、锁相环 (PLL) 时钟模块、带有实时中断 (RTI) 的看门狗定时器 (WDT) 模块等。用户可以根据各自的需求任意配置资源。

与 TMS320LF2407 器件相比, TMS320LF2407A 器件增加了以下新的特点^[2]:

- 1) 最高 40MHz 的运行速度 (2407 为 30MHz)。
- 2) 片上 Flash 中的代码可以加密。
- 3) PDPINTx, CAPn, XINTn 和 ADCSOC 引脚有输入保护电路。
- 4) PDPINTx 管脚的状态反映在 COMCONx 寄存器中。

TMS320LF2407 是 240x 系列 DSP 控制器中功能最强、片上设施最完备的一个型号,被广泛用于代码开发、系统仿真以及实际系统中。

16 位的定点 DSP TMS320LF2407A @40MHz 方便实现电机控制。本系统使用的片上外设和存储器资源以及外扩的资源如图 1 所示。

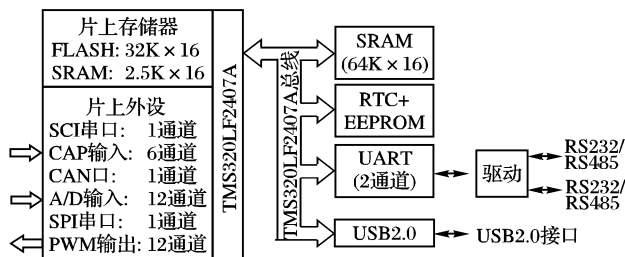


图 2 DSP 的片上资源和外扩资源示意图

3 $\mu\text{C}/\text{OS-II}$ 在 TMS320LF2407A 上的移植与裁剪

$\mu\text{C}/\text{OS-II}$ 源代码的绝大部分都是用 ANSI C 编写的,但 $\mu\text{C}/\text{OS-II}$ 是为嵌入式应用编写的通用软件,不能照顾到不同 CPU 的各自特点,只能将用汇编语言编写的与 CPU 硬件相关的部分压缩到最小。用户需要根据不同 CPU 的特点进行 $\mu\text{C}/\text{OS-II}$ 的移植工作。另外,要在处理器上移植 $\mu\text{C}/\text{OS-II}$, 处理器必须满足以下要求^[3]:

- 1) 处理器的 C 编译器能产生可重入代码;
- 2) 用 C 语言可以打开和关闭中断;
- 3) 处理器支持中断,并且能产生定时中断 (通常为 10Hz ~ 100Hz);
- 4) 处理器硬件能容纳一定数量的任务堆栈 (可能是几千字节);
- 5) 处理器有将堆栈指针和其他 CPU 寄存器读出和存储到堆栈或内存中的指令。

TMS320LF2407A 完全满足上述条件,所以系统的移植没有问题。其移植只与 4 个文件相关,即汇编文件 (OS_CPU_A.ASM)、处理器相关 C 文件 (OS_CPU.H、OS_CPU_C.C) 和配置文件 (OS_CFG.H)。系统移植所进行的工作主要有以下几个方面:

1) 使用 C 语言开关中断和软中断用于任务切换,其宏定义如下:

```
#define OS_ENTER_CRITICAL() asm(" SETC INTM");
#define OS_EXIT_CRITICAL() asm(" CLRC INTM");
#define OS_TASK_SW() asm(" INTR 31");
```

2) 用汇编语言编写几个与处理器相关的函数: OSStartHighRdy(), OSCtxSw(), OSIntCtxSw(), OSTickISR(), 在本次系统移植中, OSTickISR() 可以直接用 C 语言编写,所以没有使用汇编语言。

3) 编写 OS_CPU_C.C

OSTaskStkInit() 负责初始化任务的堆栈结构,使之看起来就像刚发生过中断并且所有必要的寄存器都已保存到了堆栈中的情况一样。执行一条 TRAP 指令 CPU 是按一定的顺序将寄存器存入堆栈的,用户在初始化堆栈的时候也就必须依照这一顺序。TMS320LF2407A 的堆栈增长方向是由低地址向高地址递增的,所以常量 OS_STK_GROWTH 必须设置为 0。

4) 时钟节拍函数 OSTickISR():

$\mu\text{C}/\text{OS-II}$ 要求用户提供一个时钟资源以实现延时功能。TMS320LF2407A 是一种先进的 DSP 芯片,内部有 4 个通用的硬件定时器。作者在移植时采用通用的硬件定时器 1 为 $\mu\text{C}/\text{OS-II}$ 提供时钟定时中断信号,其时钟节拍在定时器驱动部分设定。

OSTickISR() 的主要代码如下:

```
interrupt void OSTickISR( void)
{
    switch( PIVR)
    {
        case 0x0027:
            OSIntEnter();
            OS_EXIT_CRITICAL();
            //用户代码
            OSTimeTick(); /* 时间片中断服务 */
            EVAIFRA = EVAIFRA&0x0080;
            OSIntExit();
            break;
            //用户代码
        default:
            break;
    }
    return;
}
```

OSStartHighRdy(), OSCtxSw(), OSIntCtxSw() 这几个函数由 TMS320LF2407A 对应的汇编语言编写,这里不作详细描述。做完上述工作后,移植的工作即已完成,在时钟中断处理过程中,作者又嵌入了一级高优先级中断。该高优先级中断处理完成后,系统能回到低优先级中断继续运行,这证明了系统能够实现中断的嵌套,符合一般实时操作系统的要求。

$\mu\text{C}/\text{OS-II}$ 是可以裁剪的。为了使程序代码的执行效率高、占用空间小、实时性好,用户必须对该操作系统进行裁剪和配置。OS_CFG.H 文件中包含系统主要的配置参数。为了用户裁剪和配置方便,建立了操作系统配置表 (表 1)。该配置表是 Excel 表格,具有自动计算功能,例如 OS_TCB 占用的字节数 (表 1 中字段 #Bytes) 的计算公式为:

$$3 * \text{POINTER} + 6 * \text{INT8U} + \text{INT16U} + (\text{BOOLEAN} * \text{OS_TASK_DEL_EN}) + (\text{POINTER} * \text{OS_Q_EN} * \text{OS_MBOX_EN}) + (\text{POINTER} * \text{OS_MBOX_EN} * \text{OS_Q_EN} * \text{OS_MUTEX_EN} * \text{OS_SEM_EN}) + (2 * \text{POINTER} + \text{INT32U} + 2 * \text{INT16U}) * \text{OS_TASK_CREATE_EXT_EN}$$

OS_Q_EN 所需程序存储器空间字节数(表 1 中字段 CODE Bytes)的计算公式为:

$$\text{OS_Q_EN} * (1460 + \text{OS_Q_DEL_EN} * 341 + \text{OS_ARG_CHK_EN} * 265)$$

表 1 $\mu\text{C}/\text{OS-II}$ 操作系统配置表

Configuration Parameters	Value	CODE DATA	
		/Byte	/Byte
OS_MAX_EVENTS	6	70	
OS_MAX_MEM_PART	1	24	
OS_MAX_QS	1	28	
OS_MAX_TASKS	20	550	
OS_LOWEST_PRIO	20	87	
OS_TASK_IDLE_STK_SIZE	512	1024	
OS_TASK_STAT_EN	1	260	10
OS_TASK_STAT_STK_SIZE	512	1024	
OS_ARG_CHK_EN	1		
OS_CPU_HOOKS_EN	1		
OS_MBOX_EN	1	875	
OS_MBOX_DEL_EN	0		
OS_MEM_EN	1	841	
OS_MUTEX_EN	0	0	
OS_MUTEX_DEL_EN	0		
OS_Q_EN	1	1725	
OS_Q_DEL_EN	0		
OS_SEM_EN	1	739	
OS_SEM_DEL_EN	0		
OS_TASK_CHANGE_PRIO_EN	1	470	
OS_TASK_CREATE_EN	1	300	
OS_TASK_CREATE_EXT_EN	0	0	
OS_TASK_DEL_EN	1	525	
OS_TASK_SUSPEND_EN	1	305	
$\mu\text{C}/\text{OS-II}$ Internals		2700	47
Total Application Stacks	0	0	
Total Application RAM	0	0	
TOTAL		8740	2864

表 2 2407A Data Sizes

Data Structures	#Bytes
Compiler Alignment	2
BOOLEAN	1
INT8S	1
INT8U	1
INT16U	2
INT32U	4
OS_STK	2
POINTER	4

NOTE: DO NOT CHANGE ANYTHING BELOW!

OS_EVENT	11
OS_MBOX_DATA	8
OS_MEM	20
OS_MEM_DATA	12
OS_Q	24
OS_Q_DATA	12
OS_SEM_DATA	6
OS_STK_DATA	8
OS_TCB	25

用户只要把系统裁剪后相应参数的值(表 1 中字段 Value)确定后,输入表 1 相应的 Excel 表中,就可以知道裁剪

后的操作系统需要程序存储器空间 ROM 和数据存储器空间 RAM 的大小。例如,本系统对 TMS320LF2407A 配置后需要 ROM 8740 个字节,需要 RAM 2864 个字节。

4 移植后的测试和应用

初步完成移植后,作者编写了优先级不同的 7 个任务,其具体定义如表 3 所示。

表 3 应用程序任务定义表

优先级	任务名	任务功能描述
0	TaskStart()	开始任务,完成系统初始化及其他任务的创建
7	TaskCanRX()	CAN 接收任务
8	TaskCanTX()	CAN 发送任务,等 CAN 接收任务释放信号量后执行
10	TaskUARTRX()	UART 接收任务
11	TaskUARTTX()	UART 发送任务,周期发送
14	TaskCAP()	捕获任务,用于频率或周期测量,周期性任务
16	TaskAD()	模拟量输入读取任务,周期性读取
18	TaskBit()	周期性自检任务,完成机内自检

除了上面所列出的 7 个任务之外,作者还编写了多个中断服务程序和任务调用的一些功能函数,使应用程序读起来简洁流畅。

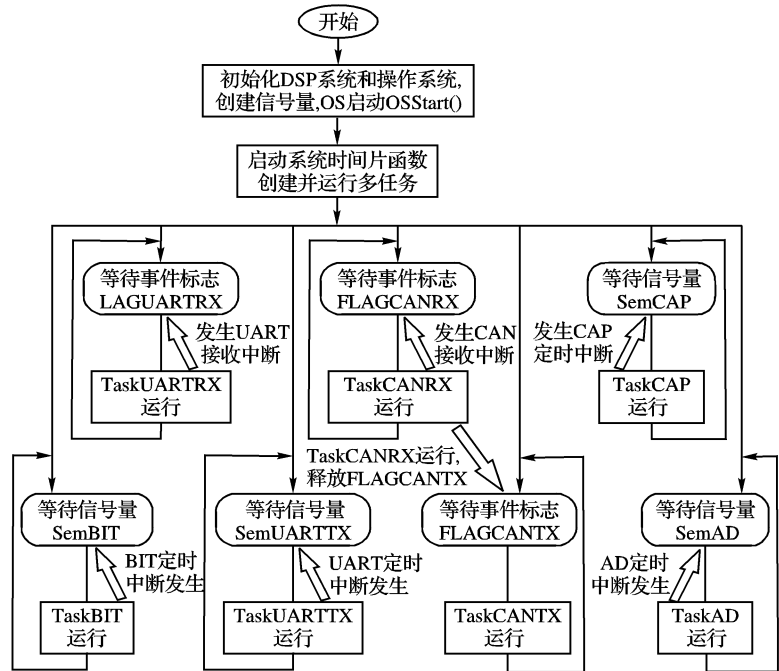


图 3 系统任务运行程序流程

通过创建和运行多任务,分别测试了如下功能函数:

- 1) 任务创建 OSSemCreate(), 任务删除 OSTaskDel();
- 2) 信号量创建 OSSemCreate(), 挂起 OSSemPend() 和释放 OSSemPost();
- 3) 事件标志创建 OSFlagCreate(), 挂起 OSFlagPend() 和释放 OSFlagPost();
- 4) 延时函数 OSTimeDly() 等。

$\mu\text{C}/\text{OS-II}$ 内核中主要功能函数的执行情况,运行均正常。同时作者还在每一个任务中都通过使 TMS320LF2407A 的 GPIO 的某一位翻转以点亮和熄灭该任务对应的指示灯。给任务设定不同的延时时间,任务所对应的指示灯就可按相应的频率闪烁。

利用如下的 Chebychev 度量计算两幅图像之间的距离:

$$d_{i,j} = \max_k |x_{ik} - x_{jk}| \quad (12)$$

可得到如表 2 所示的距离矩阵。

表 2 以高斯描绘子为特征的距离矩阵

模 板	旋转后的字母						伸缩后的字母					
	A	B	C	D	E	F	A	B	C	D	E	F
A	0.17	5.99	14.66	13.89	7.25	2.43	0.20	5.72	14.90	13.96	7.05	2.68
B	5.75	0.33	10.41	9.54	4.10	6.71	5.86	0.08	10.56	9.61	4.29	6.66
C	14.64	10.64	0.20	2.40	14.57	17.18	14.70	10.52	0.19	2.39	14.76	17.12
D	13.72	9.24	2.39	0.06	13.64	16.26	13.77	9.60	2.35	0.13	13.84	16.20
E	7.34	4.46	14.47	13.70	0.27	4.83	7.46	4.10	14.72	13.77	0.20	4.57
F	2.71	7.15	17.16	16.39	4.62	0.20	2.83	6.79	17.41	16.46	4.42	0.19

为方便对比,我们再计算轮廓矩不变量,结果如表 3 所示(因为原值太小,我们对其绝对值取常用对数)。

表 3 英文字母的轮廓矩不变量

字母	样式	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7
A	模板	-2.44	-6.56	-7.44	-9.26	-18.72	-12.83	-17.61
	旋转	-2.44	-6.54	-7.44	-9.27	-19.08	-12.82	-17.63
	伸缩	-2.44	-6.58	-7.43	-9.24	-18.52	-12.86	-17.58
B	模板	-2.49	-6.11	-8.69	-9.55	-18.69	-12.60	-19.16
	旋转	-2.52	-6.13	-8.77	-9.77	-19.05	-12.84	-19.77
	伸缩	-2.49	-6.12	-8.70	-9.56	-18.72	-12.62	-19.15
C	模板	-2.22	-5.56	-7.93	-8.63	-16.93	-11.42	-17.47
	旋转	-2.22	-5.56	-7.91	-8.64	-16.93	-11.42	-17.47
	伸缩	-2.22	-5.57	-7.95	-8.63	-16.95	-11.42	-17.40
D	模板	-2.33	-6.23	-8.13	-9.52	-18.34	-12.63	-20.02
	旋转	-2.33	-6.24	-8.13	-9.50	-18.32	-12.62	-19.21
	伸缩	-2.34	-6.24	-8.14	-9.49	-18.31	-12.61	-19.25
E	模板	-2.40	-5.63	-9.26	-10.53	-21.50	-13.42	-20.43
	旋转	-2.45	-5.73	-9.33	-10.92	-21.85	-13.84	-21.05
	伸缩	-2.40	-5.63	-9.30	-10.50	-20.84	-13.46	-20.43
F	模板	-2.28	-5.12	-7.44	-8.35	-16.30	-10.94	-16.59
	旋转	-2.33	-5.22	-7.58	-8.49	-16.56	-11.12	-16.91
	伸缩	-2.28	-5.12	-7.43	-8.34	-16.28	-10.93	-16.56

依然用 Chebychev 度量计算两幅图像之间的距离,得到表 4。

表 4 以轮廓矩不变量为特征的距离矩阵

模 板	旋转后的字母						伸缩后的字母					
	A	B	C	D	E	F	A	B	C	D	E	F
A	0.37	2.16	1.79	1.61	3.44	2.15	0.20	1.54	1.77	1.64	2.81	2.43
B	1.54	0.61	1.76	0.56	3.17	2.25	1.59	0.03	1.76	0.55	2.15	2.61
C	2.15	2.31	0.02	1.75	4.92	0.56	1.59	1.79	0.07	1.79	3.91	0.91
D	2.40	0.71	2.56	0.80	3.52	3.12	2.45	0.87	2.63	0.77	2.50	3.47
E	2.80	2.45	4.57	3.18	0.63	4.93	2.98	2.78	4.55	3.19	0.66	5.22
F	2.41	3.18	0.88	2.63	5.55	0.32	2.22	2.56	0.81	2.67	4.54	0.03

(上接第 2761 页)

5 结语

$\mu\text{C}/\text{OS}-\text{II}$ 移植到 TMS320LF2407A 上,实现了电压电流信号的采集、频率的测量、CAN 通信等多个功能,为系统进一步功能扩展和应用到电力系统测控中打下基础。

参考文献:

按最小距离分类,由表 2 可知,高斯描绘子的识别率为 100%;由表 4 可知,旋转和伸缩后的字母 D 都被误识为 B,所以轮廓矩不变量的识别率仅为 $10/12 = 83\%$ 。而且,从这两个表还可看出,相对于轮廓矩不变量,高斯描绘子的类间距离更大,类内距离更小。

我们还用很多其他字符做过实验,结果表明:高斯描绘子的识别率通常都大于 95%,而轮廓矩不变量的识别率接近 87%;至于类间和类内距离,结论与上述相同。

3 结语

本文将基于边缘的形状特征高斯描绘子用于字符识别,并与轮廓矩不变量(也是一种基于边缘的特征,是经典的 Hu 矩的推广和改进)进行比较。实验结果表明,较之后者,高斯描绘子具有高得多的识别率,且类内距离很小,类间距离很大。

参考文献:

- [1] CHELLAPPA R, BAGDAZIAN R. Fourier coding of image boundaries[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984, 6(1): 102-105.
- [2] SEKITA I, TORAICHI K, MORI R, et al. Feature extraction of handwritten Japanese characters by spline functions for relaxation matching[J]. Pattern Recognition, 1988, 21(1): 9-17.
- [3] TAKAHASHI H. A neural net OCR using geometrical and zonal pattern features[A]. Proceedings of First International Conference on Document Analysis and Recognition [C]. Saint-Malo, France, 1991. 821-828.
- [4] KIMURA F, SHRIDHAR M. Handwritten numerical recognition based on multiple algorithms[J]. Pattern Recognition, 1991, 24(10): 969-983.
- [5] EL-KHALY F, SID-AHMED MA. Machine recognition of optically captured machine printed arabic text[J]. Pattern Recognition, 1990, 23(11): 1207-1214.
- [6] TSIRIKOLIAS K, MERTZIOS BC. Statistical pattern recognition using efficient two-dimensional moments with applications to character recognition[J]. Pattern Recognition, 1993, 26(6): 877-882.
- [7] KHOTANZAD A, HONG YH. Invariant image recognition by Zernike moments[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12(5): 489-497.
- [8] KASTURI R, TRIVEDI MM. Image Analysis Applications[M]. New York: Marcel Dekker, 1990.
- [9] ANDREWS HC. Multidimensional rotations in feature selection[J]. IEEE Transactions on Computers, 1971, 20(9): 1045-1051.
- [10] 刘亦书,杨力华,孙倩. 轮廓矩不变量及其在物体识别中的应用[J]. 中国图像图形学报, 2004, 9(3): 308-313.
- [11] MALLAT S, HWANG WL. Singularity detection and processing with wavelets[J]. IEEE Transactions on Information Theory, 1992, 38(2): 617-643.

- [1] 刘和平,王维俊,江渝,等. TMS320LF240x DSP C 语言开发应用[M]. 北京:北京航空航天大学出版社, 2003.
- [2] TI. TMS320LF/LC240XA DSP Controllers System and Peripherals Reference Guide[Z], 2001.
- [3] LABROSSE JL. 嵌入式实时操作系统 $\mu\text{C}/\text{OS}-\text{II}$ [M]. 第 2 版. 邵贝贝,等译. 北京:北京航空航天大学出版社, 2003.
- [4] 尹勇,欧光军,关荣锋. DSP 集成开发环境 CCS 开发指南[M]. 北京:北京航空航天大学出版社, 2003.