



10年口碑积累，成功培养50000多名研发工程师，铸就专业品牌形象

华清远见的企业理念是不仅要良心教育、做专业教育，更要受人尊敬的职业教育。

《从实践中学 ARM 体系结构与接口技术》

作者：华清远见

专业始于专注 卓识源于远见

第 6 章 GPIO 编程

本章简介

GPIO 控制介绍是接口技术中最简单的一种。本章通过介绍 S3C2410X 芯片的 GPIO 控制方法，让读者初步掌握控制硬件接口的方法。本章主要内容：

- GPIO 功能介绍
- S3C2410X 芯片的 GPIO 控制器详解
- S3C2410X GPIO 的应用

6.1 GPIO 功能介绍



首先应该理解什么是 GPIO。GPIO (General-Purpose IO ports) 即通用 IO 口。在嵌入式系统中常常有数量众多, 但是结构却比较简单的外部设备/电路, 对这些设备/电路, 有的需要 CPU 为之提供控制手段, 有的则需要被 CPU 用做输入信号。而且, 许多这样的设备/电路只要求一位, 即只要有开/关两种状态就够了, 比如控制某个 LED 灯亮与灭; 或者通过获取某个管脚的电平属性来达到判断外围设备的状态。对这些设备/电路的控制, 使用传统的串行口或并行口都不合适, 所以在微控制器芯片上一般都会提供一个“通用可编程 IO 接口”, 即 GPIO。接口至少有两个寄存器, 即“通用 IO 控制寄存器”与“通用 IO 数据寄存器”。数据寄存器的各位都直接引到芯片外部, 而对这种寄存器中每一位的作用, 即每一位的信号流通方向, 则可以通过控制寄存器中对应位独立地加以设置, 比如可以设置某个管脚的属性为输入、输出或其他特殊功能。

在实际的 MCU 中, GPIO 是有多种形式的。比如, 有的数据寄存器可以按照位寻址, 有些却不能按照位寻址, 这在编程时就要加以区分。比如传统的 8051 系列, 就区分成可位寻址和不可位寻址两种寄存器。另外, 为了使用方便, 很多 MCU 的 GPIO 接口除去两个标准寄存器必须具备外, 还提供上拉寄存器, 可以设置 IO 的输出模式是高阻, 还是带上拉的电平输出, 或者不带上拉的电平输出。这使得在电路设计中, 外围电路就可以简化不少。

6.2 S3C2410X 芯片的 GPIO 控制器详解



S3C2410X 芯片上共有 71 个多功能的输入/输出引脚, 它们分为 7 组 I/O 端口。

- (1) 1 个 23 位的输出端口 (端口 A)。
- (2) 2 个 11 位的输入/输出端口 (端口 B、H)。
- (3) 4 个 16 位的输入/输出端口 (端口 C、D、E、G)。
- (4) 1 个 8 位的输入/输出端口 (端口 F)。

在运行程序之前必须对每个用到的引脚功能进行设置, 如果某些引脚的复用功能没有使用, 可以先将该引脚设置为 I/O 端口。

6.2.1 S3C2410X GPIO 常用寄存器分类

(1) 端口控制寄存器 (GPAICON-GPHCON)。在 S3C2410X 中, 大多数的引脚都复用, 所以必须对每个引脚进行配置。端口控制寄存器 (GPnCON) 定义了每个引脚的功能。

(2) 端口数据寄存器 (GPADAT-GPHDAT)。如果端口被配置成了输出端口, 可以向 GPnDAT 的相应位写数据。如果端口被配置成了输入端口, 可以从 GPnDAT 的相应位读出数据。

(3) 端口上拉寄存器 (GPBUP-GPHUP)。端口上拉寄存器控制了每个端口组的上拉电阻的允许/禁止。如果某一位为 0, 相应的上拉电阻被允许; 如果是 1, 相应的上拉电阻被禁止。

如果端口的上拉电阻被允许, 无论在哪种状态 (INPUT、OUTPUT、DATAn、EINTn 等) 下, 上拉电阻都起作用。

6.2.2 S3C2410X I/O 口常用寄存器详解

端口 A 寄存器 (GPAICON/GPADAT) 描述如表 6-1 所示。

表 6-1 端口 A 寄存器 (GPAICON/GPADAT)

寄存器	地址	读/写	描述	复位值
-----	----	-----	----	-----

GPACON	0x56000000	R/W	端口 A 配置寄存器	0x7ffff
GPADAT	0x56000004	R/W	端口 A 数据寄存器	未定义
Reserved	0x56000008	—	保留	未定义
Reserved	0x5600000C	—	保留	未定义

GPACON[22:0]中的某一位置位，设置与该位相对应的引脚为输出口，清零某位可以设置相应的引脚为功能端口，功能端口的描述如表 6-2 所示。

表 6-2 GPACON 寄存器

GPACON	位	描 述
GPA22	[22]	0 = Output 1 = nFCE
GPA21	[21]	0 = Output 1 = nRSTOUT (nRSTOUT = nRESET & nWDTRST & SW_RESET(MISCCR[16]))
GPA20	[20]	0 = Output 1 = nFRE
GPA19	[19]	0 = Output 1 = nFWE
GPA18	[18]	0 = Output 1 = ALE
GPA17	[17]	0 = Output 1 = CLE
GPA16	[16]	0 = Output 1 = nGCS5
GPA15	[15]	0 = Output 1 = nGCS4

续表

GPACON	位	描 述
GPA14	[14]	0 = Output 1 = nGCS3
GPA13	[13]	0 = Output 1 = nGCS2
GPA12	[12]	0 = Output 1 = nGCS1
GPA11	[11]	0 = Output 1 = ADDR26
GPA10	[10]	0 = Output 1 = ADDR25
GPA9	[9]	0 = Output 1 = ADDR24
GPA8	[8]	0 = Output 1 = ADDR23
GPA7	[7]	0 = Output 1 = ADDR22
GPA6	[6]	0 = Output 1 = ADDR21
GPA5	[5]	0 = Output 1 = ADDR20
GPA4	[4]	0 = Output 1 = ADDR19
GPA3	[3]	0 = Output 1 = ADDR18
GPA2	[2]	0 = Output 1 = ADDR17
GPA1	[1]	0 = Output 1 = ADDR16
GPA0	[0]	0 = Output 1 = ADDR0

GPADAT 寄存器描述如表 6-3 所示。

表 6-3 GPADAT 寄存器

GPADAT	位	描 述
GPA[22:0]	[22:0]	当端口被配置成输出时，引脚状态和相应的位状态一致；当端口被配置成功能脚时，读出的值不一定

端口 B 寄存器 (GPBCON,GPBDAT,GPBUP) 描述如表 6-4 所示。

表 6-4 端口 B 寄存器 (GPBCON,GPBDAT,GPBUP)

寄 存 器	地 址	读/写	描 述	复 位 值
GPBCON	0x56000010	R/W	端口 B 配置寄存器	0x0
GPBDAT	0x56000014	R/W	端口 B 数据寄存器	未定义
GPBUP	0x56000018	R/W	端口 B 上拉控制寄存器	0x0

Reserved	0x5600001C	—	保留	未定义
----------	------------	---	----	-----

GPBCON 寄存器描述如表 6-5 所示。

表 6-5 GPBCON 寄存器

GPBCON	位	描 述
GPB[10]	[21:20]	00 = Input 01 = Output 10 = nXDREQ0 11 = reserved
GPB[9]	[19:18]	00 = Input 01 = Output 10 = nXDACK0 11 = reserved
GPB[8]	[17:16]	00 = Input 01 = Output 10 = nXDREQ1 11 = Reserved
GPB[7]	[15:14]	00 = Input 01 = Output 10 = nXDACK1 11 = Reserved

续表

GPBCON	位	描 述
GPB[6]	[13:12]	00 = Input 01 = Output 10 = nXBREQ 11 = reserved
GPB[5]	[11:10]	00 = Input 01 = Output 10 = nXBACK 11 = reserved
GPB[4]	[9:8]	00 = Input 01 = Output 10 = TCLK0 11 = reserved
GPB[3]	[7:6]	00 = Input 01 = Output 10 = TOUT3 11 = reserved
GPB[2]	[5:4]	00 = Input 01 = Output 10 = TOUT2 11 = reserved]
GPB[1]	[3:2]	00 = Input 01 = Output 10 = TOUT1 11 = reserved
GPB[0]	[1:0]	00 = Input 01 = Output 10 = TOUT0 11 = reserved

GPBDAT 寄存器描述如表 6-6 所示。

表 6-6 GPBDAT 寄存器

GPBDAT	位	描 述
GPB[10:0]	[10:0]	当端口被配置成输入时，外部数据可以从对应的端口中读出；当端口被配置成输出时，写入该寄存器的数据会被送到对应的引脚上；当端口被配置成功能脚时，此寄存器未定义

GPBUP 寄存器描述如表 6-7 所示。

表 6-7 GPBUP 寄存器

GPBUP	位	描 述
GPB[10:0]	[10:0]	0: 使能对应引脚的上拉功能 1: 对应引脚的上拉功能无效

端口 C 寄存器 (GPCCON, GPCDAT, GPCUP) 描述如表 6-8 所示。

表 6-8 端口 C 寄存器 (GPCCON, GPCDAT, GPCUP)

寄 存 器	地 址	读/写	描 述	复 位 值
GPCCON	0x56000020	R/W	端口 C 配置寄存器	0x0
GPCDAT	0x56000024	R/W	端口 C 数据寄存器	未定义
GPCUP	0x56000028	R/W	端口 C 上拉控制寄存器	0x0
Reserved	0x5600002C	—	保留	未定义

GPCCON 寄存器描述如表 6-9 所示。

表 6-9 GPCCON 寄存器

GPCCON	位	描 述
GPC[15]	[31:30]	00 = Input 01 = Output 10 = VD[7] 11 = Reserved
GPC[14]	[29:28]	00 = Input 01 = Output 10 = VD[6] 11 = Reserved
GPC[13]	[27:26]	00 = Input 01 = Output 10 = VD[5] 11 = Reserved
GPC[12]	[25:24]	00 = Input 01 = Output 10 = VD[4] 11 = Reserved
GPC[11]	[23:22]	00 = Input 01 = Output 10 = VD[3] 11 = Reserved
GPC[10]	[21:20]	00 = Input 01 = Output 10 = VD[2] 11 = Reserved

GPC[9]	[19:18]	00 = Input 01 = Output 10 = VD[1] 11 = Reserved
GPC[8]	[17:16]	00 = Input 01 = Output 10 = VD[0] 11 = Reserved
GPC[7]	[15:14]	00 = Input 01 = Output 10 = LCDVF2 11 = Reserved
GPC[6]	[13:12]	00 = Input 01 = Output 10 = LCDVF1 11 = Reserved
GPC[5]	[11:10]	00 = Input 01 = Output 10 = LCDVF0 11 = Reserved
GPC[4]	[9:8]	00 = Input 01 = Output 10 = VM 11 = Reserved
GPC[3]	[7:6]	00 = Input 01 = Output 10 = VFRAME 11 = Reserved
GPC[2]	[5:4]	00 = Input 01 = Output 10 = VLINE 11 = Reserved
GPC[1]	[3:2]	00 = Input 01 = Output 10 = VCLK 11 = Reserved
GPC[0]	[1:0]	00 = Input 01 = Output 10 = LEND 11 = Reserved

续表

GPCDAT 寄存器描述如表 6-10 所示。

表 6-10 GPCDAT 寄存器

GPCDAT	位	描 述
GPC[15:0]	[15:0]	当端口被配置成输入时，外部数据可以从对应的端口中读出；当端口被配置成输出时，写入该寄存器的数据会被送到对应的引脚上；当端口被配置成功能脚时，此寄存器未定义

GPCUP 寄存器描述如表 6-11 所示。

表 6-11 GPCUP 寄存器

GPCUP	位	描 述
GPC[15:0]	[15:0]	0: 使能对应引脚的上拉功能 1: 对应引脚的上拉功能无效

端口 D 寄存器 (GPDCON, GPDDAT, GPDUP) 描述如表 6-12 所示。

表 6-12 端口 D 寄存器 (GPDCON, GPDDAT, GPDUP)

寄 存 器	地 址	读/写	描 述	复 位 值
GPDCON	0x56000030	R/W	端口 D 配置寄存器	0x0
GPDDAT	0x56000034	R/W	端口 D 数据寄存器	未定义
GPDUP	0x56000038	R/W	端口 D 上拉控制寄存器	0xF000
Reserved	0x5600003C	—	保留	未定义

GPDCON 寄存器描述如表 6-13 所示。

表 6-13 GPDCON 寄存器

GPDCON	位	描 述
GPD[15]	[31:30]	00 = Input 01 = Output 10 = VD23 11 = nSS0
GPD[14]	[29:28]	00 = Input 01 = Output 10 = VD22 11 = nSS1
GPD[13]	[27:26]	00 = Input 01 = Output 10 = VD21 11 = Reserved
GPD[12]	[25:24]	00 = Input 01 = Output 10 = VD20 11 = Reserved
GPD[11]	[23:22]	00 = Input 01 = Output 10 = VD19 11 = Reserved

续表

GPDCON	位	描 述
GPD[10]	[21:20]	00 = Input 01 = Output 10 = VD18 11 = Reserved
GPD[9]	[19:18]	00 = Input 01 = Output 10 = VD17 11 = Reserved
GPD[8]	[17:16]	00 = Input 01 = Output 10 = VD16 11 = Reserved
GPD[7]	[15:14]	00 = Input 01 = Output 10 = VD15 11 = Reserved

GPDP[6]	[13:12]	00 = Input 01 = Output 10 = VD14 11 = Reserved
GPDP[5]	[11:10]	00 = Input 01 = Output 10 = VD13 11 = Reserved
GPDP[4]	[9:8]	00 = Input 01 = Output 10 = VD12 11 = Reserved
GPDP[3]	[7:6]	00 = Input 01 = Output 10 = VD11 11 = Reserved
GPDP[2]	[5:4]	00 = Input 01 = Output 10 = VD10 11 = Reserved
GPDP[1]	[3:2]	00 = Input 01 = Output 10 = VD9 11 = Reserved
GPDP[0]	[1:0]	00 = Input 01 = Output 10 = VD8 11 = Reserved

GPDDAT 寄存器描述如表 6-14 所示。

表 6-14 GPDDAT 寄存器

GPDDAT	位	描 述
GPDP[15:0]	[15:0]	当端口被配置成输入时，外部数据可以从对应的端口中读出；当端口被配置成输出时，写入该寄存器的数据会被送到对应的引脚上；当端口被配置成功能脚时，此寄存器未定义

GPDDUP 寄存器描述如表 6-15 所示。

表 6-15 GPDDUP 寄存器

GPDDUP	位	描 述
GPDP[15:0]	[15:0]	0: 使能对应引脚的上拉功能 1: 对应引脚的上拉功能无效

端口 E 寄存器（GPECON,GPEDAT,GPEUP）描述如表 6-16 所示。

表 6-16 端口 E 寄存器（GPECON,GPEDAT,GPEUP）

寄 存 器	地 址	读/写	描 述	复 位 值
GPECON	0x56000040	R/W	端口 E 配置寄存器	0x0
GPEDAT	0x56000044	R/W	端口 E 数据寄存器	未定义
GPEUP	0x56000048	R/W	端口 E 上拉控制寄存器	0x0
Reserved	0x5600004C	—	保留	未定义

GPECON 寄存器描述如表 6-17 所示。

表 6-17 GPECON 寄存器

GPECON	位	描 述
GPE[15]	[31:30]	00 = Input 01 = Output (open drain output) 10 = IICSDA 11 = Reserved

续表

GPECON	位	描 述
GPE[14]	[29:28]	00 = Input 01 = Output (open drain output) 10 = IICSDA 11 = Reserved
GPE[13]	[27:26]	00 = Input 01 = Output 10 = SPICK0 11 = Reserved
GPE[12]	[25:24]	00 = Input 01 = Output 10 = SPIMOSI0 11 = Reserved
GPE[11]	[23:22]	00 = Input 01 = Output 10 = SPIMISO0 11 = Reserved
GPE[10]	[21:20]	00 = Input 01 = Output 10 = SDDAT3 11 = Reserved
GPE[9]	[19:18]	00 = Input 01 = Output 10 = SDDAT2 11 = Reserved
GPE[8]	[17:16]	00 = Input 01 = Output 10 = SDDAT1 11 = Reserved
GPE[7]	[15:14]	00 = Input 01 = Output 10 = SDDAT0 11 = Reserved
GPE[6]	[13:12]	00 = Input 01 = Output 10 = SDCMD 11 = Reserved
GPE[5]	[11:10]	00 = Input 01 = Output 10 = SDCLK 11 = Reserved
GPE[4]	[9:8]	00 = Input 01 = Output 10 = I2SSDO 11 = I2SSDI

GPE[3]	[7:6]	00 = Input 01 = Output 10 = I2SSDI 11 = nSS0
GPE[2]	[5:4]	00 = Input 01 = Output 10 = CDCLK 11 = Reserved
GPE[1]	[3:2]	00 = Input 01 = Output 10 = I2SSCLK 11 = Reserved
GPE[0]	[1:0]	00 = Input 01 = Output 10 = I2SLRCK 11 = Reserved

GPEDAT 寄存器描述如表 6-18 所示。

表 6-18 GPEDAT 寄存器

GPEDAT	位	描 述
GPE[15:0]	[15:0]	当端口被配置成输入时，外部数据可以从对应的端口中读出；当端口被配置成输出时，写入该寄存器的数据会被送到对应的引脚上；当端口被配置成功能脚时，此寄存器未定义

GPEUP 寄存器描述如表 6-19 所示。

表 6-19 GPEUP 寄存器

GPEUP	位	描 述
GPE[15:0]	[15:0]	0: 使能对应引脚的上拉功能 1: 对应引脚的上拉功能无效

端口 F 寄存器（GPFCON,GPFDAT,GPFUP）描述如表 6-20 所示。

表 6-20 端口 F 寄存器（GPFCON,GPFDAT,GPFUP）

寄 存 器	地 址	读/写	描 述	复 位 值
GPFCON	0x56000050	R/W	端口 F 配置寄存器	0x0
GPFDAT	0x56000054	R/W	端口 F 数据寄存器	未定义
GPFUP	0x56000058	R/W	端口 F 上拉控制寄存器	0x0
Reserved	0x5600005C	—	保留	未定义

GPFCON 寄存器描述如表 6-21 所示。

表 6-21 GPFCON 寄存器

GPFCON	位	描 述
GPF7	[15:14]	00 = Input 01 = Output 10 = EINT7 11 = Reserved
GPF6	[13:12]	00 = Input 01 = Output 10 = EINT6 11 = Reserved
GPF5	[11:10]	00 = Input 01 = Output 10 = EINT5 11 = Reserved
GPF4	[9:8]	00 = Input 01 = Output 10 = EINT4 11 = Reserved
GPF3	[7:6]	00 = Input 01 = Output 10 = EINT3 11 = Reserved
GPF2	[5:4]	00 = Input 01 = Output 10 = EINT2 11 = Reserved
GPF1	[3:2]	00 = Input 01 = Output 10 = EINT1 11 = Reserved
GPF0	[1:0]	00 = Input 01 = Output 10 = EINT0 11 = Reserved

GPFDAT 寄存器描述如表 6-22 所示。

表 6-22 GPFDAT 寄存器

GPFDAT	位	描 述
GPF[7:0]	[7:0]	当端口被配置成输入时，外部数据可以从对应的端口中读出；当端口被配置成输出时，写入该寄存器的数据会被送到对应的引脚上；当端口被配置成功能脚时，此寄存器未定义

GPFUP 寄存器描述如表 6-23 所示。

表 6-23 GPFUP 寄存器

GPFUP	位	描 述
GPF[7:0]	[7:0]	0: 使能对应引脚的上拉功能 1: 对应引脚的上拉功能无效

端口 G 寄存器描述如表 6-24 所示。

表 6-24 端口 G 寄存器

寄存器	地址	读/写	描述	复位值
GPGCON	0x56000070	R/W	端口 G 配置寄存器	0x0
GPGDAT	0x56000074	R/W	端口 G 数据寄存器	未定义
GPGUP	0x56000078	R/W	端口 G 上拉控制寄存器	0xF800
Reserved	0x5600007C	—	保留	未定义

GPGCON 寄存器描述如表 6-25 所示。

表 6-25 GPGCON 寄存器

GPGCON	位	描述
GPG15	[31:30]	00 = Input 01 = Output 10 = EINT23 11 = nYPON
GPG14	[29:28]	00 = Input 01 = Output 10 = EINT22 11 = YMON
GPG13	[27:26]	00 = Input 01 = Output 10 = EINT21 11 = nXPON
GPG12	[25:24]	00 = Input 01 = Output 10 = EINT20 11 = XMON
GPG11	[23:22]	00 = Input 01 = Output 10 = EINT19 11 = TCLK1

续表

GPGCON	位	描述
GPG10 (5V Tolerant Input)	[21:20]	00 = Input 01 = Output 10 = EINT18 11 = Reserved
GPG9 (5V Tolerant Input)	[19:18]	00 = Input 01 = Output 10 = EINT17 11 = Reserved
GPG8 (5V Tolerant Input)	[17:16]	00 = Input 01 = Output 10 = EINT16 11 = Reserved
GPG7	[15:14]	00 = Input 01 = Output 10 = EINT15 11 = SPICLK1
GPG6	[13:12]	00 = Input 01 = Output 10 = EINT14 11 = SPIMOSI1
GPG5	[11:10]	00 = Input 01 = Output 10 = EINT13 11 = SPIMISO1
GPG4	[9:8]	00 = Input 01 = Output 10 = EINT12 11 = LCD_PWREN
GPG3	[7:6]	00 = Input 01 = Output 10 = EINT11 11 = nSS1
GPG2	[5:4]	00 = Input 01 = Output 10 = EINT10 11 = nSS0
GPG1	[3:2]	00 = Input 01 = Output 10 = EINT9 11 = Reserved
GPG0	[1:0]	00 = Input 01 = Output 10 = EINT8 11 = Reserved

GPGDAT 寄存器描述如表 6-26 所示。

表 6-26 GPGDAT 寄存器

GPGDAT	位	描述
GPG[15:0]	[15:0]	当端口被配置成输入时，外部数据可以从对应的端口中读出；当端口被配置成输出时，写入该寄存器的数据会被送到对应的引脚上；当端口被配置成功能脚时，此寄存器未定义

GPGUP 寄存器描述如表 6-27 所示。

表 6-27 GPGUP 寄存器

GPGUP	位	描述
GPG[15:0]	[15:0]	0: 使能对应引脚的上拉功能 1: 对应引脚的上拉功能无效

端口 H 寄存器 (GPHCON,GPHDAT,GPHUP) 描述如表 6-28 所示。

表 6-28 端口 H 寄存器 (GPHCON,GPHDAT,GPHUP)

寄存器	地址	读/写	描述	复位值
GPHCON	0x56000040	R/W	端口 H 配置寄存器	0x0
GPHDAT	0x56000044	R/W	端口 H 数据寄存器	未定义
GPHUP	0x56000048	R/W	端口 H 上拉控制寄存器	0x0
Reserved	0x5600004C	—	保留	未定义

GPHCON 寄存器描述如表 6-29 所示。

表 6-29 GPHCON 寄存器

GPHCON	位	描述
GPH[10]	[21:20]	00 = Input 01 = Output 10 = CLKOUT1 11 = Reserved
GPH[9]	[19:18]	00 = Input 01 = Output 10 = CLKOUT0 11 = Reserved
GPH[8]	[17:16]	00 = Input 01 = Output 10 = UEXTCLK 11 = Reserved
GPH[7]	[15:14]	00 = Input 01 = Output 10 = RXD2 11 = nCTS1
GPH[6]	[13:12]	00 = Input 01 = Output 10 = TXD2 11 = nRTS1
GPH[5]	[11:10]	00 = Input 01 = Output 10 = RXD1 11 = Reserved
GPH[4]	[9:8]	00 = Input 01 = Output 10 = TXD1 11 = Reserved
GPH[3]	[7:6]	00 = Input 01 = Output 10 = RXD0 11 = reserved
GPH[2]	[5:4]	00 = Input 01 = Output 10 = TXD0 11 = Reserved
GPH[1]	[3:2]	00 = Input 01 = Output 10 = nRTS0 11 = Reserved
GPH[0]	[1:0]	00 = Input 01 = Output 10 = nCTS0 11 = Reserved

GPHDAT 寄存器描述如表 6-30 所示。

表 6-30 GPHDAT 寄存器

GPHDAT	位	描述
GPH[10:0]	[10:0]	当端口被配置成输入时，外部数据可以从对应的端口中读出；当端口被配置成输出时，写入该寄存器的数据会被送到对应的引脚上；当端口被配置成功能脚时，此寄存器未定义

GPHUP 寄存器描述如表 6-31 所示。

表 6-31 GPHUP 寄存器

GPHUP	位	描述
GPH[10:0]	[10:0]	0: 使能对应引脚的上拉功能 1: 对应引脚的上拉功能无效

6.3 S3C2410X GPIO 的应用



通过 6.1 和 6.2 节的介绍，读者了解了 GPIO 的功能，以及 S3C2410 芯片 GPIO 控制器的配置方法。本节通过一个简单示例说明 S3C2410X GPIO 接口的应用。

示例将利用 S3C2410X 的 GPF4、GPF5、GPF6、GPF74 个 I/O 管脚控制 4 个 LED 发光二极管，使其有规律地闪烁。

6.3.1 电路连接

LED 接线原理图如图 6-1 所示。

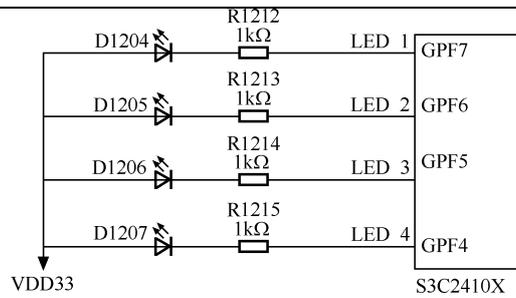


图 6-1 LED 接线原理图

如图 6-1 所示，LED₁~LED₄ 分别与 GPF7~GPF4 相连，通过 GPF7~GPF4 引脚的高低电平来控制发光二极管的亮与灭。当这几个引脚输出高电平的时候发光二极管熄灭；反之，发光二极管点亮。

6.3.2 寄存器设置

(1) 为了实现控制 LED 的目的，需要通过配置 GPFCON 寄存器将 GPF4、GPF5、GPF6、GPF7 设置为输出属性。例如，配置 GPFCON[9:8]两位为“01”，可实现将 GPF4 设置为输出属性。

(2) 通过设置 GPFDAT 寄存器实现点亮与熄灭 LED。例如，配置 GPFDAT[4]为“0”，可实现点亮 LED4；配置 GPFDAT[4]为“1”，可实现关闭 LED4。

(3) 对于本例来说，GPFUP 可以不用设置。

6.3.3 程序的编写

程序在例程目录 Chapter6\led_test 子目录下，工程名称是 led_test.Uv2，其中 LED 控制程序的核心代码如下。

(1) 相关寄存器定义。

```
#define rGPFCON    (*(volatile unsigned *)0x56000050) //端口 F 的控制寄存器
#define rGPFDAT    (*(volatile unsigned *)0x56000054) //端口 F 的数据寄存器
#define rGPFUP     (*(volatile unsigned *)0x56000058) //端口 F 的上拉控制寄存器
```

(2) 端口初始化。

```
void port_init(void)
{
    //=== PORT F GROUP
    //端口:   GPF7   GPF6   GPF5   GPF4   GPF3   GPF2   GPF1   GPF0
    //信号:   LED_1  LED_2  LED_3  LED_4  PS2_INT CPLD_INT1 KEY_INT BUT_INT1
    //设置属性: Output Output Output Output EINT3  EINT2  EINT1  EINT0
    //二进制值: 01     01,   01     01,   10     10,   10     10
    rGPFCON = 0x55aa;
    rGPFUP   = 0xff;    // GPF 所有端口都不加上拉电阻
}
```

(3) 开启 LED。

```
void led_on(void)
{
    int i,nOut;
    nOut=0xF0;
    rGPFDAT=nOut & 0x70; //点亮 LED1
    for(i=0;i<100000;i++);
    rGPFDAT=nOut & 0x30; //点亮 LED1、LED2
    for(i=0;i<100000;i++);
    rGPFDAT=nOut & 0x10; //点亮 LED1、LED2、LED3
    for(i=0;i<100000;i++);
    rGPFDAT=nOut & 0x00; //点亮 LED1、LED2、LED3、LED4
    for(i=0;i<100000;i++);
}
```

(4) 关闭 LED。

```
void led_off(void)
{
    int i,nOut;
    nOut=0;
    rGPFDAT = 0;
    for(i=0;i<100000;i++);
    rGPFDAT = nOut | 0x80; //关闭 LED1
    for(i=0;i<100000;i++);
    rGPFDAT |= nOut | 0x40; //关闭 LED2
    for(i=0;i<100000;i++);
    rGPFDAT |= nOut | 0x20; //关闭 LED3
    for(i=0;i<100000;i++);
    rGPFDAT |= nOut | 0x10; //关闭 LED4
    for(i=0;i<100000;i++);
}
```

(5) 所有 LED 交替亮灭。

```
void led_on_off(void)
{
    int i;
    rGPFDAT=0; //所有 LED 全亮
    for(i=0;i<100000;i++);
    rGPFDAT=0xF0; //所有 LED 全灭
    for(i=0;i<100000;i++);
}
```

6.4

本章小结



通过本章的学习，读者需要理解 GPIO 的概念，掌握 S3C210X 上的 GPIO 编程方法。

6.5

本章习题



1. 什么是 GPIO?
2. S3C2410X 有几组 GPIO 端口?
3. 如何实现利用 S3C2410X 的 GPD4 控制 LED? 请画出原理图，并编程实现。

联系方式

集团官网: www.hqyj.com

嵌入式学院: www.embedu.org

移动互联网学院: www.3g-edu.org

企业学院: www.farsight.com.cn

物联网学院: www.topsight.cn

研发中心: dev.hqyj.com

集团总部地址: 北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址: 北京市海淀区西三旗悦秀路北京明园大学校区, 电话: 010-82600386/5

上海地址: 上海市徐汇区漕溪路银海大厦 A 座 8 层, 电话: 021-54485127

深圳地址: 深圳市龙华新区人民北路美丽 AAA 大厦 15 层, 电话: 0755-22193762

成都地址: 成都市武侯区科华北路 99 号科华大厦 6 层, 电话: 028-85405115

南京地址：南京市白下区汉中路 185 号鸿运大厦 10 层，电话：025-86551900

武汉地址：武汉市工程大学卓刀泉校区科技孵化器大楼 8 层，电话：027-87804688

西安地址：西安市高新区高新一路 12 号创业大厦 D3 楼 5 层，电话：029-68785218

华清远见