



设计制作广告灯

1.1 学习目标

- (1) 了解 MCS-51 的资源及掌握最小系统的构成。
- (2) 了解 MCS-51 的寻址方式及指令系统。
- (3) 掌握 C51 程序设计语言。
- (4) 了解 MCS-51 程序的基本结构,掌握 C51 的简单程序设计。
- (5) 掌握程序设计软件 Keil μ Vision2、仿真软件 Proteus 及程序下载工具的使用方法。
- (6) 学会简单单片机控制系统的设计、仿真、制作与调试。

1.2 项目描述

1. 项目名称

设计制作广告灯

2. 项目要求

- (1) 用 Keil μ Vision2、Proteus 等软件作为开发工具。
- (2) 用 AT89C51 单片机控制。
- (3) 依次实现功能: 8 只 LED 灯闪烁 8 次,8 只 LED 灯奇偶交替点亮 8 次,L1~L4 与 L5~L8 交替闪烁 8 次,8 只 LED 灯全灭 1 次。
- (4) 闪烁周期时间自定。
- (5) 发挥扩充功能: 从左至右(或从右至左)轮流闪烁、拉幕功能等。

3. 设计制作任务

- (1) 拟订总体设计制作方案。
- (2) 设计硬件电路。
- (3) 编制软件流程图及设计源程序。
- (4) 仿真调试广告灯。
- (5) 安装元器件,制作广告灯,调试功能指标。
- (6) 完成项目报告。

1.3 相关知识

1.3.1 单片机简介

计算机系统向巨型化、单片化、网络化方向发展。为了提高系统的可靠性、实现微型化,把计算机系统集成在一块半导体芯片上,这种单片计算机简称单片机。它的内部硬件结构和指令系统是针对自动控制应用而设计的,所以单片机又称为微控制器 MCU(Micro Controller Unit)。经历了由4位机到8位机再到16位机的发展过程,近年来32位机已进入实用阶段。但是,由于8位机的性价比占优势,因此仍是主流机型。

单片机的制造商很多,主要有美国的 Intel、Motorola、Zilog 等公司。Intel 公司推出的 MCS-51 系列单片机使用最为广泛,拥有多种芯片,分为 51 和 52 两个子系列,如表 1-1 所示。其中 51 子系列是基本型,52 子系列是增强型。MCS-51 系列单片机中,Atmel 公司的 AT89 $\times\times$ 系列更实用:其片内存储器是 Flash 工艺,可以在线擦除、改写。对开发设备的要求低,开发时间大大缩短。国内市场 51 内核的单片机还有 STC 系列,其学习与应用也比较广泛。

表 1-1 MCS-51 单片机系列

子系列	片内 ROM 形式			ROM 容量/KB	RAM 容量/B	寻址 范围/KB	I/O 口端口			中断源
	无	ROM	EPROM				计数器	并行口	串行口	
51 系列	8031	8051	8751	4	128	2 \times 64	2 \times 16	4 \times 8	1	5
	80C31	80C51	87C51	4	128	2 \times 64	2 \times 16	4 \times 8	1	5
52 系列	8032	8052	8752	8	256	2 \times 64	3 \times 16	4 \times 8	1	6
	80C32	80C52	87C52	8	256	2 \times 64	3 \times 16	4 \times 8	1	6

单片机各个方面的性能不断提高,不仅应用于通信、网络、金融、交通、医疗、消费电子、仪器仪表、制造业控制等领域,还应用于航天、航空、军事装备领域。

1.3.2 数制与编码

1. 数制

数制也称计数制,是用一组固定的符号和统一的规则来表示数值的方法。数制中表示基本数值大小的不同数字符号称为数码。例如,十进制有 10 个数码:0、1、2、3、4、5、6、7、8、9。任何一个数制都要包含两个基本要素:基数和位权。基数是数制所使用数码的个数。例如,二进制的基数为 2;十进制的基数为 10。位权是数制中某一位上的 1 所表示数值的大小(所处位置的价值)。例如,十进制数 123,1 的位权是 100,2 的位权是 10,3 的位权是 1。

虽然计算机能极快地进行运算,但其内部并不和人类在实际生活中一样使用十进制,而是使用只包含 0 和 1 两个数值的二进制。人们输入计算机的十进制被转换成二进制进行计算,计算后的结果又由二进制转换成十进制,这都由操作系统自动完成,并不需要人们手工去做,学习单片机编程技术,就必须了解二进制、十六进制等数制。

1) 十进制数

十进制以 10 为基数,共有 0~9 十个数码,计数规律为低位向高位逢十进一。各数码在不

同位的权不一样,值不相同。例如 444,三个数码虽然都是 4,但百位的 4 表示 400,即 4×10^2 ; 十位的 4 表示 40,即 4×10^1 ; 个位的 4 表示 4,即 4×10^0 ; 其中 10^2 、 10^1 、 10^0 称为十进制数百位、十位、个位的权。一个十进制数可按每一位数展开相加,例如 585.5 可表示为

$$(585.5)_{10} = 5 \times 10^2 + 8 \times 10^1 + 5 \times 10^0 + 5 \times 10^{-1}$$

2) 二进制数

计算机中经常采用二进制。二进制的基数为 2,共有 0 和 1 两个数码,计数规律为低位向高位逢二进一。各数码在不同位的权不一样,其值不同。二进制数用下标“B”或“2”表示,如一个二进制数 101.101 按每一位数展开可表示为

$$(101.101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

3) 八进制数

在八进制数中,基数为 8。因此,在八进制数中出现的数字字符有 8 个: 0~7。每一位计数的原则为“逢八进一”,用下标“0”或“8”表示。

4) 十六进制数

在十六进制数中,基数为 16。因此,在十六进制数中出现的数字字符有 16 个: 0~9 和 A~F,其中 A、B、C、D、E、F 分别表示值 10、11、12、13、14、15。十六进制数中每一位计数原则为“逢十六进一”,用下标 H 表示。在 C51 编程中常用 0x(英文)开头表示是十六进制数,例如 0x3F。

2. 各数制之间的转换

1) R(R 表示任何数制的基数)进制数转换为十进制数

二进制、八进制和十六进制数转换为等值的十进制数,采用按权相加法。用多项式表示并在十进制下进行计算,所得的结果就是十进制数。

例如,将二进制数 1011101 转换为十进制数:

$$\begin{aligned} (1011101)_2 &= (1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0)_{10} \\ &= (64 + 0 + 16 + 8 + 4 + 0 + 1)_{10} \\ &= (93)_{10} \end{aligned}$$

2) 十进制数转换为 R 进制数

十进制数转换为等值的二进制、八进制和十六进制数,需要对整数部分和小数部分分别进行转换。其整数部分用连续除以基数 R 取余数倒排法来完成,小数部分用连续乘以基数 R 取整顺排法来实现。

例如,将十进制数 44.375 转换成二进制数(取小数点后三位)。

根据转换规则,整数部分 44 用除 2 取余倒排法:

2	44	余数	低位	0.375	整数	高位
2	22	... 0=K ₀	↑	× 2	0.750 ... 0=K ₋₁	
2	11	... 0=K ₁		0.750		
2	5	... 1=K ₂		× 2	1.500 ... 1=K ₋₂	
2	2	... 1=K ₃		0.500		
2	1	... 0=K ₄		× 2	1.000 ... 1=K ₋₃	↓
	0	... 1=K ₅	高位	1.000		低位

$$(44)_{10} = (101100)_2$$

小数部分 0.375 采用乘 2 取整顺排法:

$$(0.375)_{10} = (0.011)_2$$

所以 $(44.375)_{10} = (101100.011)_2$

3) 二进制数与八进制数、十六进制数的转换

二进制数与八进制数的转换应按“3 位二进制数对应 1 位八进制数”的原则进行;二进制数与十六进制数的转换应按“4 位二进制数对应 1 位十六进制数”的原则进行。

例如, $(101100)_2$ 转换成十六进制数:

$$(101100)_2 = (2C)_H$$

4) 二进制数的运算原则

加法:逢二进一;减法:借一当二;乘法:与算术乘法形式相同;除法:与算术除法形式相同。

3. 数据类型及数据单位

1) 数据的两种类型

计算机中的数据可分为两大类:数值型数据和字符型数据。所有的非数值型数据都要经过数字化后才能在计算机中存储和处理。

2) 数据单位

在计算机中通常使用三个数据单位:位、字节和字。位是最小的存储单位,英文名称是 bit,常用小写 b 或 bit 表示。8 位二进制数作为表示字符和数字的基本单元,称为 1 字节,英文名称是 byte。其通常用大写字母 B 表示。

$$1B(\text{字节}) = 8\text{bit}(\text{位})$$

$$1KB(\text{千字节}) = 1024B(\text{字节})$$

$$1MB(\text{兆字节}) = 1024KB(\text{千字节})$$

字长称为字或计算机字,计算机进行数据处理时,一次存取、加工和传送的数据长度称为字(word)。一个字通常由一个或多个(一般是字节的整数位)字节构成。

4. 编码

1) 8421BCD 码

用 4 位二进制数码表示 1 位十进制数,简称二-十进制码,又叫 BCD 码。其中 8421BCD 码是最常用的 BCD 码,它和 4 位自然二进制码相似,各位的权值为 8、4、2、1,与 4 位自然二进制码不同的是:它只选用了 4 位二进制码中前 10 组代码,即用 0000~1001 分别代表它所对应的十进制数 0~9,余下的六组代码不用,如表 1-2 所示。

表 1-2 8421BCD 码表

十进制数	0	1	2	3	4	5	6	7	8	9
8421 码	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

2) ASCII 码

ASCII 码使用指定的 7 位或 8 位二进制数组合来表示 128 种或 256 种可能的字符。标准 ASCII 码也叫基础 ASCII 码,使用 7 位二进制数来表示所有的大写和小写字母,数字 0~9、标点符号,以及在美式英语中使用的特殊控制字符。

0~31 及 127(共 33 个)是控制字符或通信专用字符(其余为可显示字符),如控制符:LF(换行)、CR(回车)、FF(换页)、DEL(删除)、BS(退格)、BEL(响铃)等;通信专用字符:SOH(文头)、EOT(文尾)、ACK(确认)等;ASCII 值为 8、9、10 和 13 分别转换为退格、制表、换行和回车字符。它们并没有特定的图形显示,但会依不同的应用程序而对文本显示有不同的影响。

32~126(共 95 个)是字符(32 是空格),其中 48~57 为 0~9 十个阿拉伯数字。

65~90 为 26 个大写英文字母,97~122 为 26 个小写英文字母,其余为一些标点符号、运算符号等。

在标准 ASCII 中,其最高位(b7)用作奇偶校验位。后 128 个称为扩展 ASCII 码,目前许多基于 x86 的系统都支持使用扩展(或“高”)ASCII 码。扩展 ASCII 码允许将每个字符的第 8 位用于确定附加的 128 个特殊符号字符、外来语字母和图形符号。

1.3.3 MCS-51 单片机引脚与资源

1. MCS-51 单片机的引脚

MCS-51 单片机为 40 引脚的集成芯片,其双列直插封装(DIP)形式引脚排列如图 1-1 所示。

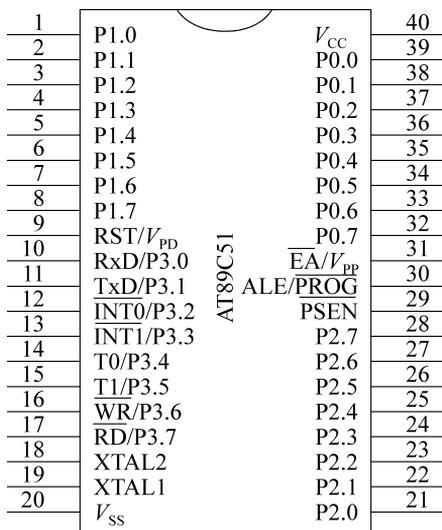


图 1-1 AT89C51 单片机引脚

1) I/O 口引脚

AT89C51 有 4 个 8 位并行 I/O 接口,共 32 条 I/O 线。

P0 口 8 条 I/O 线: P0.0~P0.7(39~32 脚)。

P1 口 8 条 I/O 线: P1.0~P1.7(1~8 脚)。

P2 口 8 条 I/O 线: P2.0~P2.7(21~28 脚)。

P3 口的 8 条 I/O 线: P3.0~P3.7(10~17 脚)。

P1、P2、P3 内置上拉电阻,P0 口需外接 10k Ω 左右的上拉电阻。P0~P3 口作输入口时,必须先写入“1”。

2) 控制信号引脚

$\overline{\text{ALE/PROG}}$ (30脚): 地址锁存允许输出信号。在系统存储器扩展时, ALE 用于控制锁存器锁存 P0 口输出的低 8 位地址, ALE 高电平期间, P0 输出地址信息, ALE 下降沿到来时, P0 口的地址信息被外接锁存器锁存, 接着出现指令和地址信息, 以实现低 8 位地址和数据的隔离。CPU 不执行访问外部存储器时, ALE 以时钟频率 1/6 为固定频率输出的正脉冲, 可作为外部时钟或外部定时脉冲使用。此引脚的第二功能是对单片机内部 EEPROM 编程时的编程脉冲输入线。

$\overline{\text{PSEN}}$ (29脚): 外部程序存储器读选通信号输出。在读外部 ROM 时 $\overline{\text{PSEN}}$ 有效(低电平), 以实现外部 ROM 单元的读操作。

$\overline{\text{EA/V}_{\text{PP}}}$ (31脚): 访问程序存储控制信号。当 $\overline{\text{EA}}$ 信号为低电平时, 对 ROM 的读操作限定在外部程序存储器; 而当 $\overline{\text{EA}}$ 信号为高电平时, 则对 ROM 的读操作是从内部程序存储器开始, 并可延至外部程序存储器。此引脚第二功能为编程电源线。

RST(9脚): 复位信号输入端, 用以完成单片机的复位操作。当单片机振荡器工作时, 连续输入 2 个机器周期以上的高电平, 单片机恢复到初始状态。

3) 外接晶振引脚

XTAL1(18脚)和 XTAL2(19脚): 外接晶振引线端。当使用芯片内部时钟时, 用于外接石英晶体和微调电容; 当使用外部时钟时, 用于接外部时钟脉冲信号。

4) 电源引脚

GND(20脚): 电源地。

V_{CC} (40脚): 电源+5V。

RST/ V_{PD} (9脚): 备用电源。

各种型号的芯片, 其引脚的第一功能信号是相同的, 不同的只在引脚的第二功能信号。

2. MCS-51 单片机的基本资源与结构

1) MCS-51 单片机基本资源

MCS-51 单片机有很多类型, 但它们基本相同。下面以 AT89C51 为例介绍单片机的内部结构。AT89C51 是 Atmel 公司推出的带有 ISP(在线编程)功能的 8 位单片机, 是目前应用的首选机型。该单片机的主要功能特性与资源如下。

完全兼容 51 系列。

工作频率 0~24MHz。

4KB Flash ROM, 并且可在线编程。

128B RAM。

32 个 I/O。

5 个中断源。

2 个 16 位定时/计数器。

1 个全双工串行通信端口。

看门狗定时器。

双数据指针。

片内时钟振荡器。

暂寄存器存放参与运算的操作数,不对外开放。

寄存器 B 是专为乘法和除法设置的寄存器,也是 8 位寄存器。乘法运算时,B 是存放乘数。乘法操作后,乘积的高 8 位存于 B 中,除法运算时,B 是存放除数。除法操作后,余数存于 B 中。此外,B 寄存器也可作为一般数据寄存器使用。

程序状态字(Program Status Word,PSW)是一个 8 位寄存器,用于储存程序运行中的各种状态信息。其中有些位状态是根据程序执行结果,由硬件自动设置的,有些位状态则使用软件方法设定。PSW 的位状态可以用专门指令进行测试,也可以用指令读出。一些条件转移指令将根据 PSW 有些位的状态,进行程序转移。PSW 的 PSW.0~PSW.7 的位地址为 D0H~D7H,各位定义如表 1-3 所示。

表 1-3 PSW 的 PSW.0~PSW.7 的含义

PSW 位地址	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H
字节地址 D0H	CY	AC	F0	RS1	RS0	OV		P

注:PSW.1 位保留未用。

CY(PSW.7)——进位标志位。表示累加器 A 在加减运算过程中其最高位 A7 有无进位或借位。如果操作结果的最高位产生进位或借位,CY 由硬件置 1,否则清 0。另外,也可以由位运算指令置位或清零。

AC(PSW.6)——辅助进位标志位。在进行加减运算中,当有低 4 位向高 4 位进位或借位时,AC 由硬件置 1,否则 AC 位被清 0。

F0(PSW.5)——用户标志位。这是一个供用户定义的标志位,根据需要可以用软件来使它置位或清除。

RS1 和 RS0(PSW.4 和 PSW.3)——寄存器组选择位。AT89C51 片内共有 4 组工作寄存器,每组 8 个,分别命名为 R0~R7。编程时用于存放数据或地址。但每组工作寄存器在内部 RAM 中的物理地址不同。RS1 和 RS0 的 4 种状态组合就是用于选择 CPU 当前使用的工作寄存器组,从而确定 R0~R7 的实际物理地址。RS1、RS0 状态与工作寄存器 R0~R7 的物理地址关系如表 1-4 所示。

表 1-4 RS1、RS0 与 R0~R7 的物理地址关系

RS1	RS0	寄存器组	片内 RAM 地址
0	0	第 0 组	00H~07H
0	1	第 1 组	08H~0FH
1	0	第 2 组	10H~17H
1	1	第 3 组	18H~1FH

这两个选择位由软件设置,被选中的寄存器组即为当前通用寄存器组。单片机上电或复位后,RS1 RS0=00。

OV(PSW.2)——溢出标志位。当执行算术指令时,由硬件自动置位或清零,表示累加器 A 的溢出状态。在带符号数运算结果超过范围(-128~+127),无符号数运算结果超过范围(255),乘法运算中积超过 255,除法运算中除数为 0,4 种情况下该位为 1。

判断该位时,通常用运算中次高位向最高位的进(借)位 C6 和最高位向前的进(借)位

C7(也就是CY)的异或来表示OV,即 $OV=C6 \oplus C7$ 。

P(PSW.0)——奇偶标志位。表明累加器A内容的奇偶性,如果A中有奇数个1,则P置1,若1的个数为偶数,则P为0。凡是改变累加器A中内容的指令均会影响P标志位。

例如,执行下列两条指令:

```
MOV A, #67H           ;将立即数送入累加器A中
ADD A, #58H           ;将A的值与立即数58H相加,结果存入A中
```

实现67H与58H相加。

67H=01100111B,58H=01011000B,加法过程为

$$\begin{array}{r} 0110\ 0111 \\ +\ 0101\ 1000 \\ \hline 1011\ 1111=0BFH \end{array}$$

执行后,A=0BFH,硬件标志位自动设置为CY=0、AC=0、OV=C6 \oplus C7,P=1,如无关位为0,则PSW=05H。

(2) 控制器

控制器是单片机内部按一定时序协调工作的控制核心,是分析和执行指令的部件。控制器主要由程序计数器(PC)、指令寄存器、指令译码器和定时控制逻辑电路等构成。

程序计数器是专门用于存放将要执行的下一条指令的16位地址,可寻址64KB范围的ROM。CPU根据PC中的地址到ROM中去读取程序指令码和数据,并送给指令寄存器进行分析。PC的内容具有自动加1的功能,用户无法对其进行读写,只能用指令改变PC的值,实现程序的跳转。

指令寄存器用于存放CPU从ROM读出的指令操作码。

指令译码器是用于分析指令操作的部件,指令操作码经译码后产生相应的信号。

定时控制逻辑电路用来产生脉冲序列和多节拍脉冲。

(3) 寄存器

寄存器是单片机内部的临时存放单元或固定用途单元,包括通用寄存器组和专用寄存器组。通用寄存器组用于存放运算过程中的地址和数据,专用寄存器用于存放特定的操作数,指示当前指令的存放地址和指令运行的状态等,51单片机共有4组32个通用寄存器,21个专用寄存器。对于特殊功能寄存器,前面介绍了累加器A、寄存器B和标志寄存器PSW,下面介绍数据指针(DPTR)和堆栈指针(Stack Pointer,SP),其余的在后面项目中介绍。

数据指针(地址)寄存器为16位寄存器,寻址范围达64KB。它既可以按16位寄存器使用,也可以按寄存器DPH(高8位)DPL(低8位)作两个寄存器使用。DPTR专门用来寄存片外RAM及扩展I/O口进行数据存取时用的地址。

堆栈是一个特殊的存储区,用来暂存数据和地址,它只有一个数据进/出端口,按“先进后出”的原则存取数据。堆栈的底部叫栈底,数据的进出口叫栈顶,栈顶的地址叫堆栈指针,用8位寄存器SP来存放,系统复位后SP的内容为07H,但是一般把堆栈开辟在内部RAM的30H~7FH单元中,空栈时栈底的地址等于栈顶的地址。

数据进入堆栈的操作叫进栈,首先SP的内容加1送入SP,然后再向堆栈存储器写入数据。

数据读出堆栈的操作叫出栈,堆栈存储器读出数据,然后SP的内容减1送入SP。

4) 存储器结构

MCS-51 单片机的芯片内部有 RAM 和 ROM 存储器,外部可以扩展 RAM 和 ROM,在物理上分为 4 个空间。逻辑上分为程序存储器(内、外统一编址,使用 MOVC 指令访问)、内部数据存储器(使用 MOV 指令访问)和外部数据存储器(使用 MOVX 指令访问)。

(1) 内部数据存储器 RAM

对于普通 8051 单片机,内部 RAM 有 256B,用于存放程序执行过程的各种变量及临时数据。低 128B 可用直接寻址或间接寻址方式进行访问,分为工作寄存器区、位寻址区、用户区和堆栈区 4 个区域,高 128B 为特殊功能寄存器区,其配置如图 1-3 所示。

① 工作寄存器区。00H~1FH 地址单元,共有 4 组寄存器,每组 8 个寄存单元(均为 8 位),都以 R0~R7 作寄存单元编号。寄存器常用于存放操作数及中间结果等,在任一时刻,CPU 只能使用其中的一组寄存器,由程序状态字寄存器 PSW 中 RS1、RS0 位的状态组合来选择,正在使用的那组寄存器称为当前寄存器组。

② 位寻址区。20H~2FH 单元为位寻址区,既可作为一般 RAM 单元进行字节操作,也可以对单元中每一位进行位操作。位寻址区共有 16 个 RAM 单元,计 128 位,位地址为 00H~7FH,如表 1-5 所示。

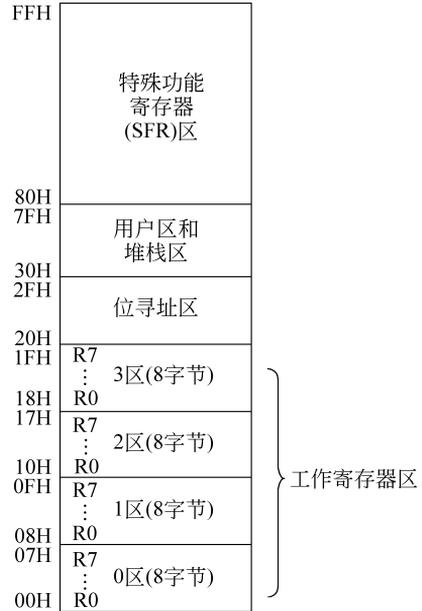


图 1-3 片内 RAM 的配置

表 1-5 位寻址区

位地址/位名称								字节地址
D7	D6	D5	D4	D3	D2	D1	D0	
7F	7E	7D	7C	7B	7A	79	78	2FH
77	76	75	74	73	72	71	70	2EH
6F	6E	6D	6C	6B	6A	69	68	2DH
67	66	65	64	63	62	61	60	2CH
5F	5E	5D	5C	5B	5A	59	58	2BH
57	56	55	54	53	52	51	50	2AH
4F	4E	4D	4C	4B	4A	49	48	29H
47	46	45	44	43	42	41	40	28H
3F	3E	3D	3C	3B	3A	39	38	27H
37	36	35	34	33	32	31	30	26H
2F	2E	2D	2C	2B	2A	29	28	25H
27	26	25	24	23	22	21	20	24H
1F	1E	1D	1C	1B	1A	19	18	23H
17	16	15	14	13	12	11	10	22H
0F	0E	0D	0C	0B	0A	09	08	21H
07	06	05	04	03	02	01	00	20H