

HT1632C 在点阵式 LED 显示屏中的应用

文件编码: HA0127S

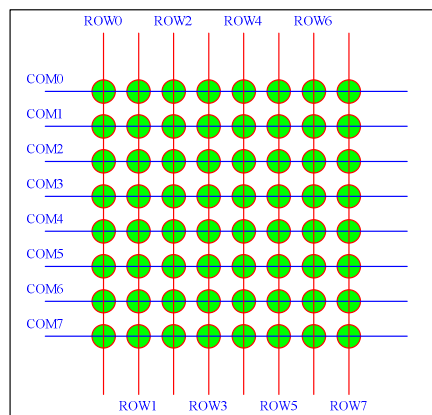
简介

HT1632C 是一种 Memory Mapping LED 显示驱动器, 它广泛应用于数字时钟、温度计、湿度计等工业仪器仪表显示中, 多片 HT1632C CASCADE 的使用方法在点阵式 LED 显示应用中尤为显著。HT1632C 的基本特性如下:

- HT1632C 工作电压: 2.4V~5.5V
- 多种显示方式 (32 ROW & 8 COM 或 24 ROW & 16 COM)
- 内建显示 RAM (若选 32 ROW & 8 COM 显示方式, 显示 RAM 为 64x4bit;
若选 24 ROW & 16 COM 显示方式, 显示 RAM 为 96x4bit)。
- 16 级 PWM 亮度控制
- 内建 256K RC 振荡器
- 与 MCU 接口为串行接口方式
- 与 MCU 通讯有命令指令和数据指令
- COM 线可选为 N-MOS Open Drain 输出驱动或 P-MOS Open drain 输出驱动

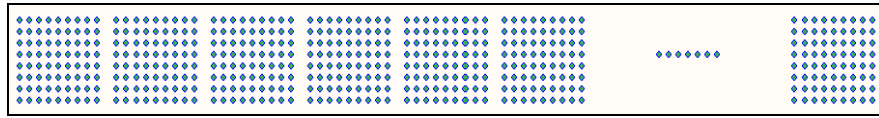
HT1632C 驱动 LED 点数评估

点阵式 LED 显示屏结构



最小的 LED 点阵板

我们现在要用 HT1632C 驱动的 LED 显示屏是用 18 块如上图所示的 LED 点阵板组成的 144×8 的点阵屏幕，效果图如下：



HT1632C 芯片选取评估

上面已经介绍了一片 HT1632C 有 32×8 或 24×16 两种显示方式，我们现在要推动 144×8 点 LED 显示屏，HT1632C 应选 24×16 显示方式，这样可驱动的 LED 点数较多。 $(144 \times 8) / (24 \times 16) = 3$ ，这样用 3 片 HT1632C 就可完成以上 LED 显示屏的驱动。

单片 HT1632C 可驱动 $24 \times 16 = 384$ 点 LED，一块 LED 点阵板有 64 点，那么一片 HT1632C 则可驱动 6 块最小的 LED 点阵板。3 片 HT1632C 则可驱动 18 块 LED 点阵板。

再如，我们要点亮的 LED 点阵显示屏点数更大，那么对 HT1632C 芯片的数量需求就越多。

推动 HT1632C 芯片的 MCU 选取

MCU 选取要根据自己所选图形数量多少及图形显示变化方式如何来确定 MCU 的资源，图形数量较多时会占用较多的 ROM 空间，图形显示方式变化较多时会占用较多的 RAM 空间。HT1632C 只需占用 MCU 的 3 根输出线。

HT1632C 驱动 LED 点阵屏图形显示方法

图形显示概述

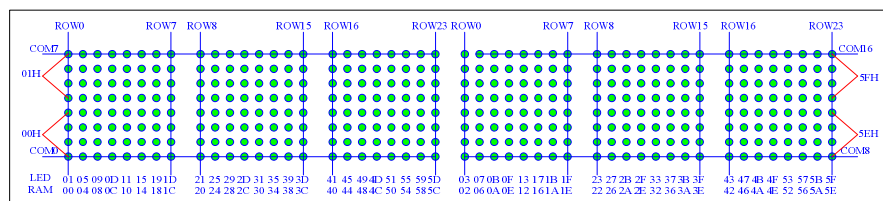
HT1632C 是一种 Memory Mapping LED 显示驱动器，这种驱动方式为图形显示带来了极大便利。实际上利用 Memory Mapping LED 驱动器显示图形就变成了对 LED RAM 的一种逻辑写入操作。

在 LED 点阵式显示屏中，我们一律将要显示的对象称为图形，例如：“H” 单个字符，我们将其看作一幅图形，再如“HOLTEK SEMICONDUCTOR INC.”，也同样看作图形处理。

每一幅图在我们 MCU 的 Memory 中均有一组数据与之对应，HT1632C 的图形显示就是将这些数据写入到 LED 驱动器的 RAM 中。

LED 点阵中的点与 LED RAM 的对应关系

在下图中我们给出了单片 HT1632C 驱动 6 个 LED 块时，LED 点阵中的点与 LED RAM 的对应关系，一般情况下在显示时，将此 LED RAM 建成一个表，然后根据位置指针来从表中取出 LED RAM，再将相应的图形数据写到 LED RAM 中，图形即可显示出来。



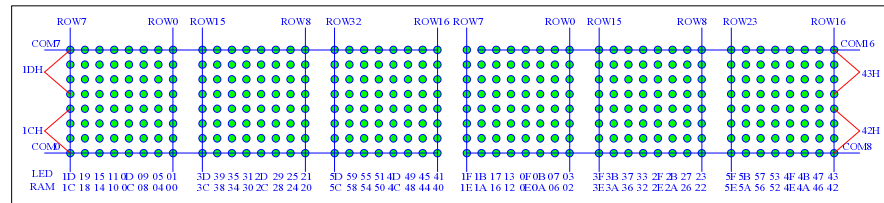
LED_RAM_TABLE:

DC 00H, 01H, 04H, 05H, 08H, 09H, 0CH, 0DH, 10H, 11H, 14H, 15H, 18H, 19H,
1CH, 1DH

DC 20H, 21H, 24H, 25H, 28H, 29H, 2CH, 2DH, 30H, 31H, 34H, 35H, 38H, 39H,
3CH, 3DH

.....

若 HT1632C 的 ROW 线和 COM 线与 LED 点阵块的连线发生了变化, 如下图, 则只需修改 LED 表中的 LED RAM 顺序, 而不需修改程序。可见利用查表方法是比较方便的。



LED_RAM_TABLE:

DC 1CH, 1DH, 18H, 19H, 14H, 15H, 10H, 11H, 0CH, 0DH, 08H, 09H, 04H, 05H,
00H, 01H

DC 3CH, 3DH, 38H, 39H, 34H, 35H, 30H, 31H, 2CH, 2DH, 28H, 29H, 24H, 25H,
20H, 21H

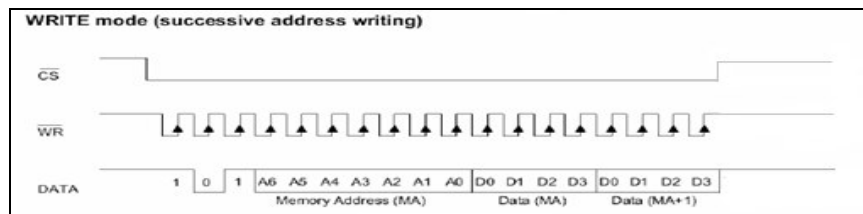
.....

图形数据的建立

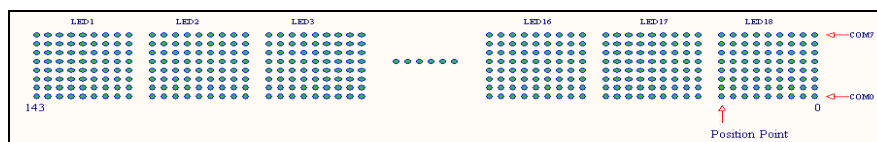
任何一幅图形都是以数据形式存放在 MCU 的 Memory 中, 图形数据的建立, 一般尽量与 LED RAM 及 LED 点阵中的点相一致, 例如: COM7, ROW0 的交叉点, 对应地址 01H 的 bit3; COM4, ROW0 的交叉点, 对应地址 01H 的 bit0; COM3, ROW0 的交叉点, 对应地址 00H 的 bit3; COM0, ROW0 的交叉点, 对应地址 00H 的 bit0; 01H, 00H 两个地址共 8bit, 因此在建立图形数据时, 最高位 bit7, 要对应 01H 的 bit3, 数据的最低位 bit0, 要对应 00H 的 bit0。如此建立图形数据, 将来数据写到 LED RAM 中就很方便。

如何写数据到 HT1632C 中

图形数据在 MCU 的 Memory 中的存放是以 8bit 格式存放的, 但 LED RAM 中的每个地址为 4bit, 如何将一个 8bit 数据放到两个 4bit RAM 中, 我们可以利用 HT1632C 的连续位地址写入模式, 当然要写入 8 bit 数据, 只需连续写两个 RAM 地址就可以了。如下图 (注意: 写入位地址时先送出高电平, 写入数据时先送出低电平)。



144×8 大屏幕位图形显示方法



我们的LED点阵显示屏实际上是一个如上图所示的144×8的一个点阵块，那么我们现在如何在上面LED点阵显示屏中显示一幅图形，例如“HOLTEK”，其方法为：首先我们要知道该图形要显示的位置，如上图，位置坐标从右到左排列为0~143，共144列。我们设定一个位置指针(Position Point)记录目前要显示的位置。另外，我们再设定一个取数据指针(Data Point)，我们根据位置指针(Position Point)求出(查表或计算)该列所对应的LED RAM的地址，再根据取数据指针(Data Point)取出该列所对应的图形数据，将取出的数据写到相应的LED RAM中，则该列就显示出来，要显示下一列，位置指针和数据指针均要加1，如此重复直到一个图形数据全部显示完毕。则在上图LED点阵显示屏的任一位置就显示出一幅图形。

图形在屏幕中移动方法

- 图形从右向左移动

图形移动的原理就是：先擦掉已有的图形，然后改变图形显示的位置指针，再在新的位置上显示新图形。如此以一定时间间隔显示图形过程就会产生图形在屏幕中移动的效果。图形从右向左移动，每经过一个时间间隔，位置指针(Position Point)应加1，当然位置指针增加，要显示图形的大小也在变化，位置指针的大小加上1，也就是要显示图形(列数)的大小。如此按照一定时间间隔增加位置指针(Position Point)，直到位置指针(Position Point)指向143列。
- 图形从下向上移动

图形从下向上移动与图形从右向左移动有所不同，其方法为每次将数据写入RAM之前，要对数据进行处理，例如在某列中要显示的数据为：10011001B，则图形从下向上移动时，第一次先将数据10011001B右移7bit，右移时高电平填“0”，数据变为0000001B，将此数据写到LED RAM中，在图形显示中每个数据均如此处理，图形就会在屏幕的最下面一行显示出来。要显示屏幕下面的两行内容，则对数据的处理只需将原始数据右移6bit，右移时高电平填“0”，则数据变为0000010B，将此数据写到LED RAM中，就可将屏幕的下面两行内容显示出来。当图形移到最顶端时，则不需再移动数据直接将数据写到LED RAM即可。
- 其它图形变化显示方法

图形在LED点阵显示屏中的变化显示是丰富多彩的，更多的图形变化显示方法可参考我们下面范例例程。

HT1632C 应用范例介绍

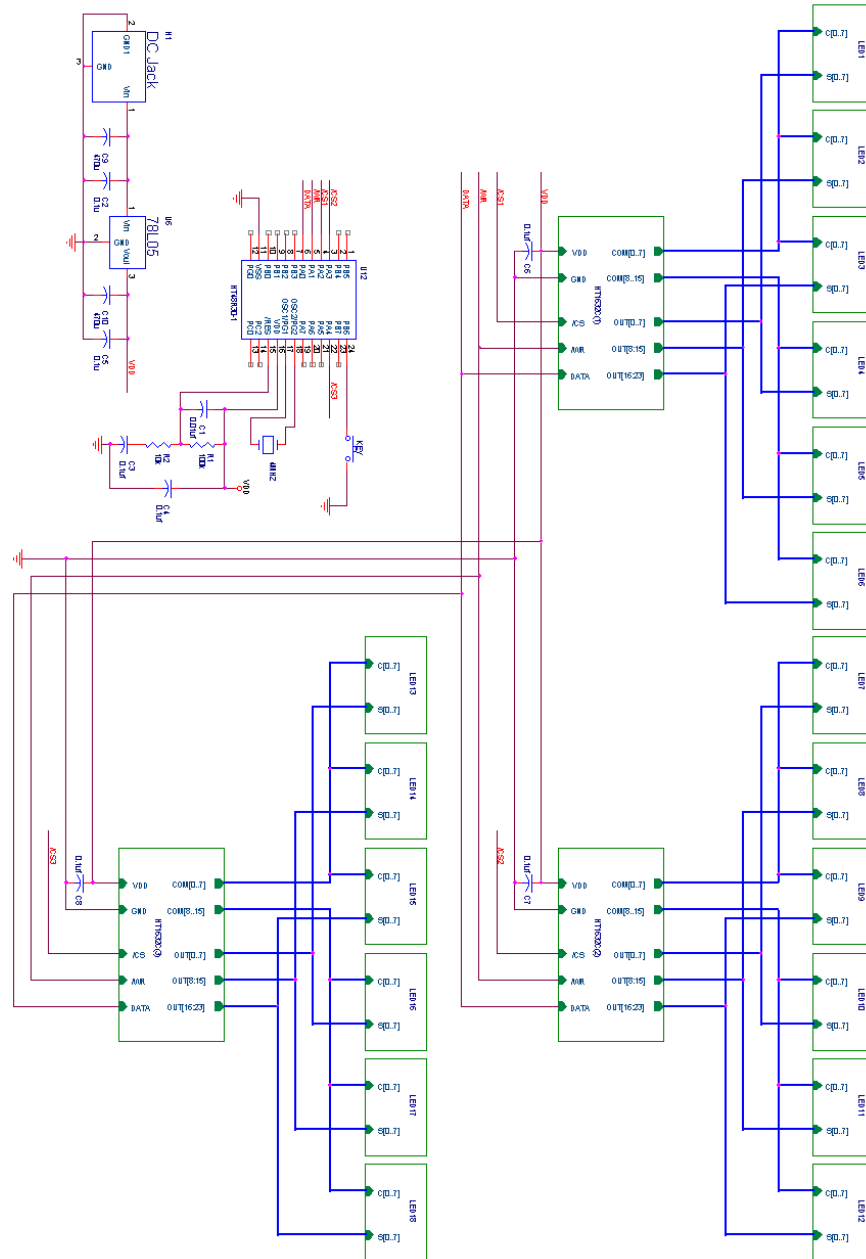
根据上述对 HT1632C 应用的讲解，下面我们给出一个具体实例来加深对 HT1632C 应用的理解。此范例用 Holtek C 语言完成，MCU 用 HT48R30A-1。

DEMO 程序所展示的功能

此 DEMO 程序是在以上 144×8 LED 点阵屏中完成以下图形变化显示，来展示 HT1632C 推动 LED 的能力。

- 将“HOLTEK SEMICONDUCTOR INC.”图形，从右到左逐渐显示到屏幕中间。
- 对“HOLTEK SEMICONDUCTOR INC.”图形，进行闪烁。
- 将“HOLTEK SEMICONDUCTOR INC.”图形，向左逐渐移出屏幕。
- 将“HOLTEK SEMICONDUCTOR INC.”图形，从上下两边显示到屏幕中间。
- 将“HOLTEK SEMICONDUCTOR INC.”图形，从中间往上下两边逐渐灭掉。
- 将“HOLTEK SEMICONDUCTOR INC.”图形，从左右两边显示到屏幕中间。
- 将“HOLTEK SEMICONDUCTOR INC.”图形，从中间往左右两边逐渐灭掉。
- 显示 HOLTEK 及其 LOG 标识。

DEMO 电路图



- 电源部分。
Demo Board 用 DC9V 供电, 经 LM7805 稳压后, 提供 5V 电压给 HT48R30A-1 和 HT1632C。
- HT48R30A-1 与 HT1632C 接口。
HT48R30A-1 与 HT1632C 接口共三根线: 写线 (WR)、数据线 (DATA)、选线 (CS)。
- HT1632C 电源和地之间的去耦电容。
由于此显示屏较大, 电源线和地线到每个 HT1632C 芯片的走线比较长, 所以每个 HT1632C 的 VDD 和 VSS 之间一定要接一个 104 去耦电容, 如 C6、C7、C8。
- MCU 振荡电路及 RESET 电路部分。
 1. Y1 为晶振振荡电路。
 2. C1、C3、C4、R1、R2 组成 Power On Reset 电路。

DEMO 软件说明

- 文件结构
 - 在此 DEMO 程序中共包含 4 个档，即 main.c、sub.c、1632_driver.c 和 isr_tmr.c。其中 sub.c、1632_driver.c、isr_tmr.c 三个档用 INCLUDE 方式包含在 main.c 中。
 - 在 main.c 中，主要完成了对 HT48R30A-1 MCU 的初始化（RAM、I/O PORT、TIMER 设定和中断设定等），同时也对 HT1632C 进行了初始化设定。
 - 在 isr_tmr.c 中，主要是 Timer 中断服务函数。
 - 在 1632_driver.c 中，主要是 HT48R30A-1 推动 HT1632C 的各种命令函数。
 - 在 sub.c 中主要包含图形变换显示的一些函数。
- 主要变量说明
 - M_TMR_MS
Timer 中断时间间隔为 8ms，用此变量进行 1 秒钟分频。
 - M_TMR_SEC
1 秒钟将该变数加“1”一次，用作秒计数器。
 - M_DATA_SPEED
8ms 将此变数加“1”一次。
 - M_Function_step
图形显示功能计数器，当完成一个功能后此变量加“1”，再完成下一个功能。所有功能显示完后此变量归“0”，又从第一功能开始演示。
- 主要函数说明
 - Main()函数
Main()函数是程序开始的主函数。
 - Ini_Memory()函数
该函数主要用来初始化 HT48R30A-1 的 RAM。
 - Ini_System()函数
该函数主要用来设置 HT48R30A-1 的 I/O 口、Timer、中断等。
 - Ini_1632()函数
该函数主要用来初始化 HT1632C。初始化过程如下：
 1. 向 HT1632C 写入“100”，设为命令模式。
 2. 向 HT1632C 写入“0x01”，Turn On HT1632C System Clock。
 3. 向 HT1632C 写入“0x2c”，COM 线选为 P-MOS 输出，COM 线个数选为 16COM。
 4. 向 HT1632C 写入“0x03”，LED On。
 5. 向 HT1632C 写入“0x08”，Set HT1632C Blink Off。
 6. 向 HT1632C 写入“0xaa”，LED 亮度 Duty Circle 设为 10/16。
 7. HT1632C 的全部 RAM 全部清除为“0”。
 这里要注意：在对 HT1632C 进行初始化时，可以将 3 片 HT1632C 的片选信号 CS 同时 Enable，3 片 HT1632C 可同时进行初始化。
 - SBR_DATA_DisplayCS()函数
该函数主要功能是一片 HT1632C 驱动 6 个 LED 点阵块函数，通过此函数的调用，可在 6 个 LED 块中显示出一幅图形。该函数入口有两个参数，CSEn 及 Station，CSEn 用来指明那一片 HT1632C 显示图形，Station 用来指明图形显示的位置。
 - SBR_DATA_DisplayByte()函数
该函数主要功能是一片 HT1632C 驱动 6 个 LED 点阵块函数，通过此函数的调用，可在 6 个 LED 块中显示出一列图形。该函数入口有三个参数，CSEn、Station 及 Data，CSEn 用来指明哪一片 HT1632C 显示图形，Station 用来指明图形显示的位置，Data 是要显示的数据。
 - SBR_FUNCTION_STEP0()函数
该函数主要功能完成图形从右向左移入屏幕功能。
 - SBR_FUNCTION_STEP2()函数
该函数主要功能是对屏幕中的图形进行闪烁，闪烁方法为发 LED ON 命令到 HT1632C 或发 LED OFF 命令到 HT1632C。
 - SBR_FUNCTION_STEP4()函数

该函数主要功能是图形逐渐向左移出屏幕。

- BR_FUNCTION_STEP6()函数
该函数主要功能是从上下两边开始，逐渐向中间点亮 LED 显示屏。
- BR_FUNCTION_STEP8()函数
该函数主要功能是从中间向上下两边逐渐灭掉 LED 显示屏。
- BR_FUNCTION_STEP10()函数
该函数主要功能是从左右两边开始逐渐向中间逐渐点亮 LED 显示屏。
- SBR_FUNCTION_STEP12()函数
该函数主要功能是从 LED 屏幕中间开始逐渐向两边熄灭 LED 显示。
- BR_FUNCTION_STEP14()函数
该函数主要功能是左边 6 个最小 LED 点阵块全亮，右边 6 个最小 LED 点阵块 LED 全亮，中间 6 个 LED 点阵块灭掉。
- BR_FUNCTION_STEP16()函数
该函数主要功能是左边 6 个最小 LED 点阵块全亮，右边 6 个最小 LED 点阵块全亮，中间 6 个 LED 点阵块显示“HOLTEK”及 LOGO 字样。
- BR_FUNCTION_STEP18()函数
该函数主要功能是对 LED RAM 全部清除“0”。

总结

首先，在本文中我们介绍了 HT1632C 的基本特征以及如何用多片 HT1632C 驱动较大点阵的 LED 显示屏。

其次，我们介绍了 HT1632C 作为 Memory Mapping LED 驱动器的图形显示方法及图形在屏幕中的移动方法。

最后，我们给出一个具体范例，使大家更好的理解 HT1632C 的应用。

当然，我们此文的讲解只是抛砖引玉，希望大家能举一反三，利用更多的 HT1632C 芯片扩展成更大的 LED 点阵显示屏幕，显示更丰富的内容。

参考文献

HT1632C, HT48R30A-1 说明书

(可从以下网站下载：<http://www.holtek.com.cn>、<http://www.holtek.com.tw>)。

版本记录

版本：V1.10

修改日期：2010 年 6 月 14 日

修改内容：产品型号由 HT1632 更改为 HT1632C 及硬件电路图更新。