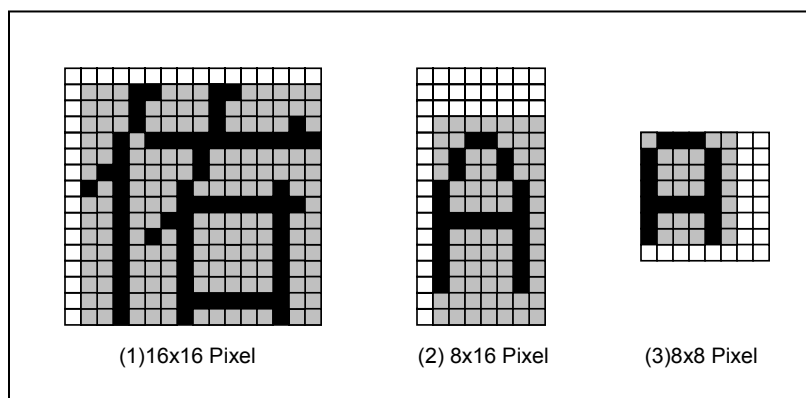


RA8815 内建 256KByte 的字型 ROM，储存着中文字型、数字符号、英、日、欧、拉丁文等字型，因为中文字型的不同，RA8815 提供两种型号：

1. **RA8815G-T**：代表内建的中文字型为繁体字
2. **RA8815G-S**：代表内建的中文字型为简体字

因此在开发及采购时必须先确定使用的 RA8815 型号。而 RA8815 内建的字号共有三种：

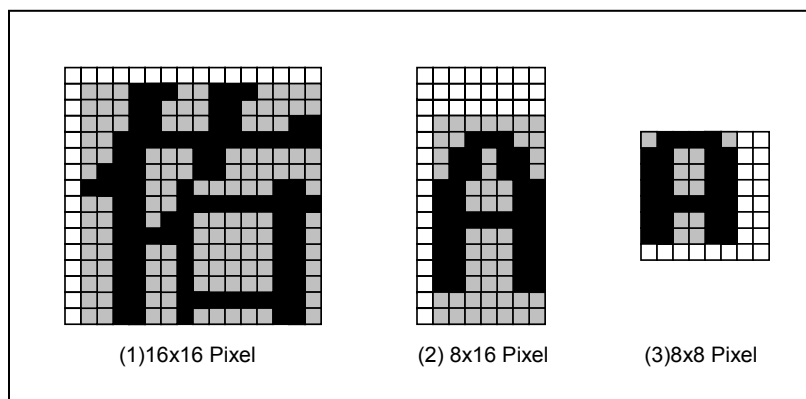
1. **中文字型**：所占的显示区域大小为 16x16，实际字体大小为 15x15，如图 1 的(1)，所支持的字收录在 RA8815 应用手册“附录 D. 字型与字码表”，其实由字码表中可以看出所包含的字不只中文，还有包括一些符号、数字等，而这些字型在 RA8815G-T 与 RA8815G-S 是不一样的，RA8815G-T 字型的对映码是 BIG5 码，RA8815G-S 字型的对映码是 BG 码，如使用 RA8815G-T，当 MCU 在设定好坐标位置后送出 Data “B7h”及”E7h”，就会显示”瑞”的繁体中文字型，使用 RA8815G-S，当 MCU 在设定好坐标位置后送出 Data “C6h”及”CCh”，就会显示”铺”的简体中文字型。



<图 1> 一般显示字型

2. **大 ASCII 字型**：所占的显示区域大小为 8x16，实际字体大小区域一般显示字型为 6x13，如图 1 的 (2)，粗体字显示字型为 7x13，如图 2 的(2)。大 ASCII 字型共有 1024 个字(请参考 Table-1 与 Table-2)，其中包括一些符号、数字、英、日、欧文及拉丁文，大 ASCII 字型在 RA8815G-T 与 RA8815G-S 也是一样的。只要 RA8815 设定成大 ASCII 字型模式(REG[03h]的 MD[1:0]=10)，由 MCU 送出 Table-1 或 Table-2 的字型码就会显示出所对映的 8x16 字型，如 MCU 在设定好坐标位置后送出 Data “41h”，就会显示”A”的大 ASCII 字型。选择大 ASCII 表由缓存器[03h]MWMR 的 Bi4 ASCS 决定，ASCS 为 0 选择 ASCII 表 1(Table1)，为 1 选择 ASCII 表 2(Table2)。
3. **小 ASCII 字型**：所占的显示区域大小为 8x8，实际字体大小区域一般显示字型为 5x7，如图 1 的 (3)，粗体字显示字型为 6x7，如图 2 的 (3)。小 ASCII 字型共有 512 个字(请参考 Table-0)，其中包括一些符号、数字、英、日、欧文等，小 ASCII 字型在 RA8815G-T 与 RA8815G-S 是一样的。只要 RA8815 设定成小 ASCII 字型模式(REG[03h]的 MD[1:0]=01)，由 MCU 送出 Table-0 的字型码就会显示出所对映的 8x8 字型，如 MCU 在设定好坐标位置后送出 Data “50h”，就会显示”P”的小

ASCII 字型。



<图 2>粗体字显示字型

值得注意的是 Table-0~2 这些字型所对映的码是由本公司自行规划的，虽然包括一些符号、数字、英、日、欧文及拉丁文，但所对映的字型码与英、日、欧文及拉丁文等操作系统所定义的标准码无关，因此 RA8815 不论透过 MCU 送出的是哪一种文字码或是哪一国的标准码，在大、小 ASCII 模式下，LCD 将只显示出由 RA8815 接收且对映到 Table-0~2 的字型。而如果不同的操作系统应用上您想显示的字是在 Table-0~2 内，但是操作系统产生的字型码与 Table-0~2 上的不同，那么就必须在程序上自行建立一转码表，例如您想显示大 ASCII 字型 "Ů"，在操作系统中产生的字型码却是 "A5h"，那么在程序上的转码表就是要由 "A5h" 对映到 "C8h"，唯有由 MCU 送出 "68" 才可以让 RA8815 显示出 "Ů"，又如您想显示大 ASCII 字型 "£"，在操作系统中产生的字型码却是 "9Ch"，那么在程序上的转码表就是要由 "98h" 对映到 "9Ch"，因为唯有由 MCU 送出 "9C" 才可以让 RA8815 显示出 "£"。

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
3	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
4	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
5	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
6	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
7	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
8	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
9	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
A	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
B	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
C	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
D	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
E	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
F	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

小 ASCII 表(Table 0)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
3	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
4	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
5	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
6	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
7	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
8	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
9	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
A	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
B	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
C	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
D	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
E	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
F	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

大 ASCII 表(Table 1)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
1	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
3	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
4	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
5	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
6	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
7	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
8	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
9	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
A	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
B	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
C	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
D	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
E	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
F	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

大 ASCII 表(Table 2)