

主编&美工: yls

fx-ES PLUS series 研究报告

百度FX-ES(MS)吧荣誉出品

- 字符关系
- 内存结构
- 神级应用
- 异常模式

说在前面 ..

2012 一年成果 PLUS 系列爆发的说 . 一切都是从某位高手的 991ES PLUS 按出乱码开始的 . PLUS 系列的研究达到了前所未有的深度和强度 , 甚至可以直接控制显存 ..

我们在 PLUS 系列上实现了像 ES 一样的 " 溢出 " , 而且利用 BUG 实现了各种各样近乎无法想象的效果 .

做这个东西主要是为了对已经完成的工作做一个比较系统的总结 , 同时让新来的人可以快速的达到我们现在的 " 知识水平 " , 加速入门 , 避免重复发现被说火星冥王星之类 . 以此让贴吧尽量持久的发展下去 .

这个 PDF 里 , 有关计算机术语都以通俗方式表达了 , 方便理解 . 同时本人编程无力怕有出错的地方 . .

如果对内容或者排版美工方面有什么疑问或者建议 , 可以直接 @ 本人百度 ID yls_1996 , 也可以直接到贴吧发帖 . 同时也非常希望有更多的人加入我们 [fx-es\(ms\) 吧](#)

P.s. 所有[蓝色](#)的字都可以点击直接进入相关页面 !

【Magic of CALC fx-es(ms) 吧 2012 成果总集】一定要先看的一个[视频](#)

视频和这个 PDF 里面白色壳贴了兔斯基的是本人换壳的 991+

由于排版关系 , 有一些大图无法置入 PDF , 各种对照表之类 . 我把他们都放到了一个[资料包](#)里大家自行下载

也可以到 [-PLUS 系列研究实用性资料大全 -](#)

PDF 中的一些经过剪切的资料图都可以在那里找到原图

如果你想亲自实践内存部分 , 会用到 [82ES](#) 和 [82ESPLUS](#) [绿色模拟器](#) 还会用到 WINHEX 这个软件 , 自己百度便是 . 到处都是

如果你有时间 , 兴趣和精力的话 , 建议先看一下[计算器基础功能的高级运用方法](#) . 这个东西 .. 也许在页面上浏览会有问题 , 所以最好下载

下面几个整理贴 , 几乎囊括了所有的成果 . 内容全面但是比较杂乱 , 但是也不妨去看看

[比较完整细致的 F&Q 和各类问题解决](#)

[991+ 成果整理贴](#)

[82+ 成果整理贴](#)

【PLUS 系列精品贴列表】(当做版权说明吧 ~) [百度在线阅读有问题最好下载](#)

以下内容是按照比较容易理解和把握的顺序编写的 , 内容无缝衔接 . 如果你从来没有接触过相关方面 . **请按顺序阅读**

目录

可以直接点击标题跳到相关页哟 0(n _ n)0~~

专有名词 / 按键扫盲

自检

二进制位 & 16 进制 计算机基础知识

条形码拼字 (图案 & 中文)

拼任意英文句子

利用不稳定字符刷出任意符号

控制字符 (索引)

WINHEX 修改内存方法

爆机

数据爆机

乱点模式 & 命名

异常模式

计时模式

自动复制模式

ASCII 模式 (两种)

乱点 / 异常乱码 / ES 升级的实质

模式控制

119 模式

68 模式

82+ 溢出

82+ 乱点模式

82+ 计时模式

82+BASE-N 模式

82+ 程序 / 升级 / 自检深层次讨论

82+ASCII 模式

变量储存原理 / 锁机

按键扫描 (应用)

专有名词 / 按键扫盲

82/991/95+
加号为 PLUS 的简写

线性模式

LineIO, 输入和输出都没有自然书写的形式, 在 math 下按 shift, mode, 2 进入 (参见说明书)

数学模式

MathIO, 输入输出都以自然书写形式表现, 在支持的模式下按 shift, mode, 1, 1 进入 (参见说明书)

乱点模式

顾名思义是屏幕上有乱七八糟点的模式, 是堆栈生长点移动的结果 (主要在下方, 会上涨)。乱点模式在 PLUS 内是异常模式的入口。

X 次乱点 进入乱点时共计按等于号的次数, 比如 15 下等于号为 15 乱点。进入乱点前按等于号实质上是往前移动应用堆栈的数据生长点。

异常模式 按一定数量的字符会出现乱码的模式。

自检

用于检测屏幕, 程序和按键是否正常的的一个特殊的 模式, 通常供工厂做检测机器是否正常使用

然后是一些会用到也许你不知道的按键 (991+) (其实说明书都有 = =)

CALC: 键盘左上角 shift 下面那个

输入等于号: 按 alpha, 然后 CALC (后文中的输入等于号, 即用这个输入, 按等于号就是右下角那儿)

积分号: CALC 右边那个

西格玛: 按 shift, 有两个框的 log 输入

幂: x 右上角又一个小框的键

计算器自检: 检查本 PDF 所涉及的内容是否适用于你的计算器

在 SHIFT 和 7 一起按住的情况下按 ON, 屏幕即会显示 DIAGNOSTIC Press AC. 这时按 9 就进入自检了。按几次 shift 屏幕会出现版本信息

以下内容涉及实体机操作的仅适用于自检信息为 GY455X VerE 的 991ES PLUS 和 GY450X VerE 的 82ES PLUS

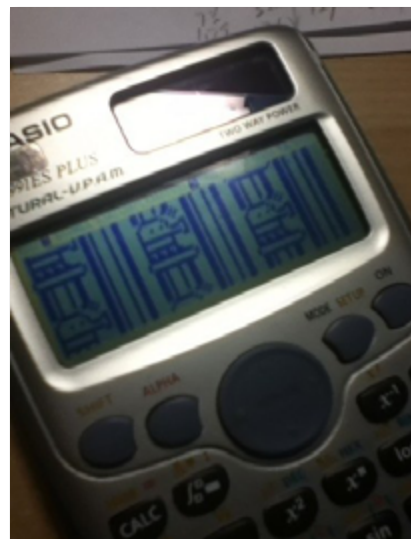
如果是 VerB 或者国内的 VerA (假货) 都是不可以的 ..

一般验证是否正版:

依次按住数字 1, 数字 0, 减号, 等于号, 然后只松开 1, 屏幕会出现 cos(的一定是正品 (四按一松法) 原理见最后一页键盘扫描

字符部分。。这一部分的应用的 PLUS 娱乐的精髓。。各种神图 所以认真看哟

首先普及一下一些基础知识。大家都知道电脑是以 0 和 1 表示数据也就是二进制 (Bin)，一个 0 或者 1 被称为一个二进制位，简称 1 位 (1bit)，然后，8bit 组成一字节 (1byte)，1byte 即用两个 16 进制 (HEX) 的数值表示。从 00 到 FF(看不懂的自己百度“字节” ‘十六进制’)



然后，ASCII 字符是组成计算器里面所有符号之类的基本字符。见图。总共有 256 个。【ASCII 的前 128 个是世界通用的(计算器的和你的电脑的手机的都一样)，后 128 个则是拓展定制的。看图就知道。每一个 ASCII 对应一个 HEX 数值。然后，再往下一级，每一个一般输入的字符(比如 sin(，加减号数字什么的) 会对应一个 ASCII 字符。实际上，Bin，Hex，ASCII 和一般的字符可以看做是数据的不同表现形式 (也就是 - 编码 -)。

fx-es PLUS系列ASCII字符表

正常输入字体 5*9

```

0123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[^\_
-abcdefghijklmno
pqrstuvwxyz[]~
!@x10w0r9/23456789
x10w0r9/23456789
0123456789-1x10()±
012-1FNPABCDEFPP
Σαγδθλμπσφχψω
,~?@()~123456789
09~1234567890123456789
PQRSTUVWXYZ[^\_

```

小字体 (菜单显示) 5*6

```

0123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[^\_
-abcdefghijklmno
pqrstuvwxyz[]~
!@x10w0r9/23456789
x10w0r9/23456789
0123456789-1x10()±
012-1FNPABCDEFPP
Σαγδθλμπσφχψω
01234567890123456789
01234567890123456789
01234567890123456789

```

R4 WU4DF2

二进制数据直接表现在屏幕上，1 表示亮，0 表示暗。八个一组，256 种组合方式
 HEX 数据一般不会直接看到，除了一些十进制形式什么的（变量储存部分）00-FF
 计算器所有的菜单都是 ASCII 字符直接组成的。包含大小写的左右字母还有所有的符号。
 一般输入的字符是最好理解的，在计算器上按什么就出什么 = =.. 这就是最普通的输入字符，
 实际有 256 个只是不能通过本身就有的按键直接把所有的都打出来

在实体机上，条形码画画或者拼字就使输入的一般字符以二进制是形式在屏幕显示出来
 拼英文句子是使输入的一般字符转换成 ASCII 显示。



好了字符部分原理就这么多，必须看懂的说。。于是你也可以制造神图了。接下来上操作方法



先是如图英文句子的操作。(991+) 括号内为提示，两步操作中用横线 (-) 隔开

开机 - 分数线 - 等于号 - AC-shift-MODE-2(进入线性模式右下角一个 0)-Alpha- 右括号 - Alpha-CALC-Shift- 积分号 - Shift- 右括号 - 数字 3(现在应该是 $X=d/dx(X,3)$)-CALC- 等于号 (现在 syntax ERROR), - 长按右键 (光标不会动了) - 任意两个数字比如 22, 然后按 5 乘号乘号, 5 乘号乘号, 循环按下去, 大约一百个的时候屏幕会出现刚才打的算式, 继续按那个式子会被覆盖, 再按一百个左右会看到一个屏幕左上的 M 会亮. 现在按 AC, 左键, 等于. 屏幕最上面会有一排 NN5NN5.....十六个; 嗯这就是刚才打的东西转换成 ASCII 了, 刚才覆盖原式第一个 X 的乘号就是这里最前面的 N, 覆盖等于号的就是这里的第二个, 依次往后推前 16 个转换结果被显示出来. 乘号对应 ASCII 的是 N. 具体对应关系参见资料包的对照表. 附录有 26 个字母的 (大小写)

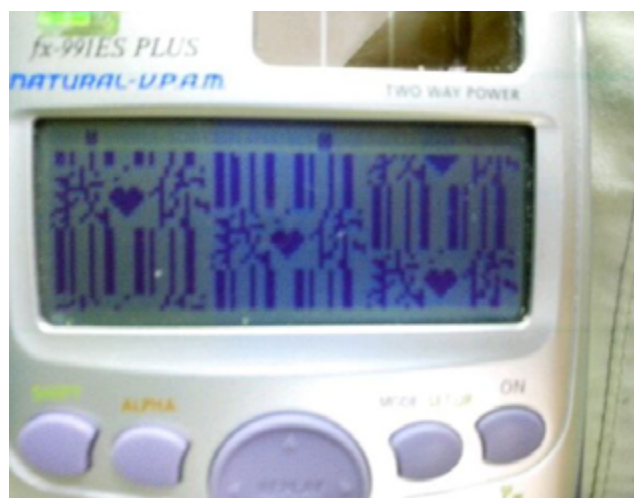
举个例子。第一次出现原式的时候按下面的指示覆盖 $X = dx(X, 3$

$$X = dx/x(X , 3$$

$\text{Rec}(\Sigma(\text{Rec}(e^{\wedge}(e^{\wedge}(\text{二次方}(\text{根号右边那个}))$

然后再随便按十个字符（加上前面打的十六个）再按乘号，5，乘号乘号5，不停的按到 M 亮，AC 左等于。你懂的

除了这种方式，还有一种方式也可以部分实现普通字符到 ASCII 的翻译，我们叫他 ASCII 模式。那个模式只能打出部分字母但是优势在于它是正常输入大小的。如左图

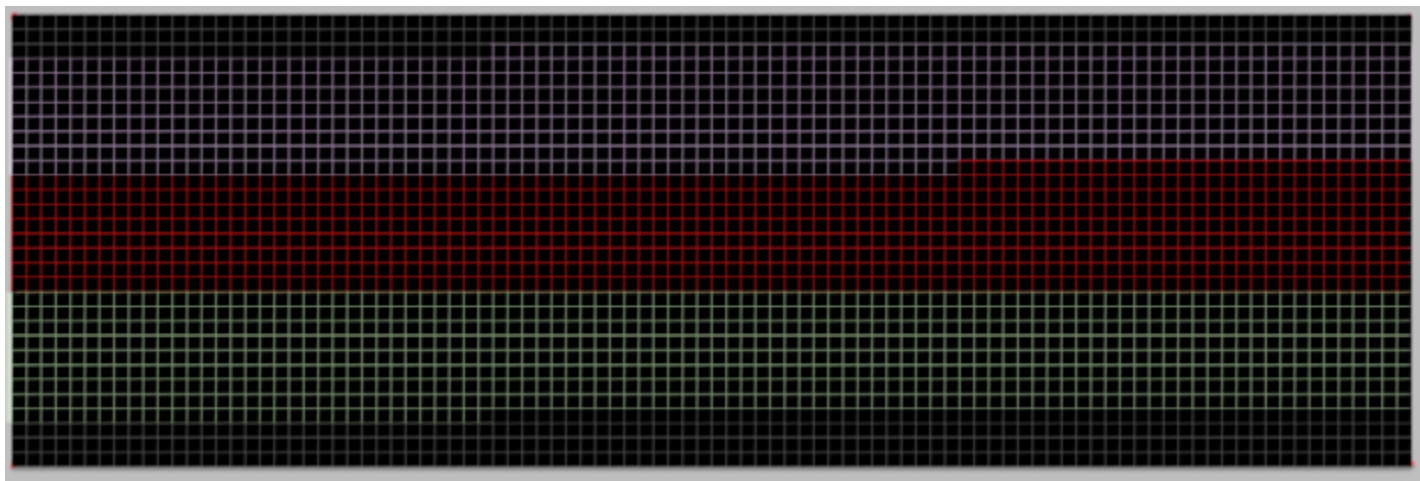


条形码拼字（右图）

具体操作

先到刚才长按右键光标卡住的地方。随便按什么的按到出原式，以覆盖第一个 X 的算第一个，第 53, 54, 55 个分别打 $\log()$ ，根号， $\log()$ 。然后随便按什么按到 M 亮，AC，左，等于，MODE，MODE。你会看到整个屏幕都是花的。so，刚才那 100 个里面只有提到的三个是必须固定的。他们作为把输入的数据转为显示的标志存在，其他的可以任意操纵用来拼东西。

前面说了一字节是 8bit 是数据，在显示上也就是一字节控制屏幕上的八个点。（不懂的进制转换关系自己百度学 = =）。计算器的屏幕是 96*31, 也就是一行 12 个字节。画画拼字什么实际上是把 9 打的东西放到显存里面显示出来。图片里可以看到图象是有重复的，因为只有覆盖第一个 X 的那个开始算往后 100 个字符的数据被循环放到显存。如下图。相同颜色的是一个循环的单位。



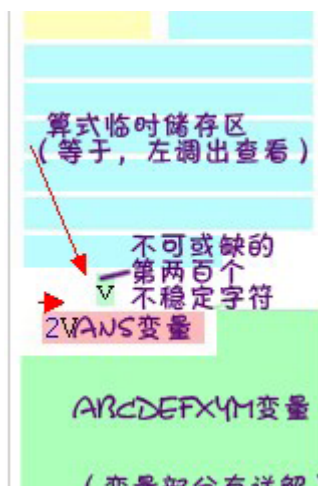
有一种比较直观的观察方法。从光标卡住那里开始，按根号根号 Ans 根号根号 Ans 到 M 亮，AC 左等于 MODE。可以看到屏幕有不是平滑衔接的地方。那里就是两个循环之间的节点

示例图片里面的图案大小明显比一个循环可以控制的范围大。也是利用了循环这种现象。具体的自己试试就知道了。说清楚需要很多字 = =|||

如果不太懂的继续往下看内存部分，用模拟器改改显存就懂了。或者，[显存控制原理原帖](#)

利用不稳定字符：

然后还有一个小问题，前面的操作中，假如你想刷的字母或者是图案需要不能直接打出来的一般字符怎么办呢？于是现在就要利用第两百个的【不稳定字符】（M 亮的前十个左右）看到那个地方，光标移动的时候那个字符会自己变。它是按照每个一般字符对应的 HEX 编号从小到大变化的（规律，这是规律）时间的话，光标自然状态从亮到灭或者从灭到亮那个字符便会自动往后跳一个。当你有操作是時候变化结果就会显示出来。利用这个，先在那里填入离你想要字符最近的那个可以打出来的，然后按时间规律左右移动光标等他一个一个变成你想要的，然后快速删掉它前面的于是这个字符就被固定下来了。

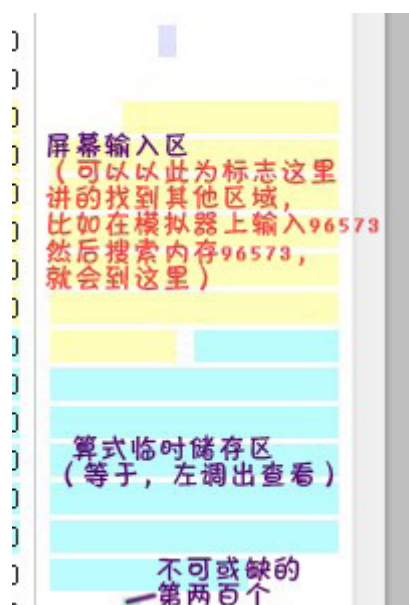


以上字符这些内容都是。自己试一下就很简单很容易的说 ..

然后到内存部分

内存修改器的使用方法

我用的是 WINHEX : 同时打开模拟器和 WINHEX , 在 WINHEX 的工具栏点工具 - 查看内存 - 选择模拟器的进程 - 全部内存 (英文版的 E 文自己看) 。然后在模拟器上随便输入一些数字 , 比如 99665 , 然后在 WINHEX 搜索内存 99665 , 就可以到输入区。修改时候必须滑动鼠标滚轮刷新内存模拟器才会有反应 .WINHEX 强烈建议用英文版。中文版右边的 ASCII 很多都是中文乱七八糟的。



前面那些, 拼图案啊英文啊什么的都输入了超过计算器本身限制的 100 个字符 . 计算器一般输入的地方确实最多只能装 100 个字符, 那多打的那一百多个肯定是被放到别的不应该的地方去了 . 于是看图 . 黄色标记的是一般的输入区, 最多容纳 100 个, 多打的那些字符实际上是打到紧接着输入区的临时储存区去了 . 水蓝色部分 . ([内存完整结构图](#))

然后, 我们知道操作计算器输入东西然后按等于号后它总会执行一些命令, 首先检查输入算式的合法性, 不合法就报错, 合法的话就按你输入的符号什么的进行计算然后输出答案 .. 但是, 如

果输入的东西超过了程序本身设计的范围, 就会发生一系列奇妙的事情 .so. 这就是最最核心的东西, 想要知道在这种状态下程序是如何异常运行的必须有程序的源代码 .. 当然, 我们没有 = =|||. 尽管没有, 我们也是可以在这个范围的逻辑下进行一些探索, 比如前面说的两个的神图制造 .

然后继续, 虽然计算器在异常状态下做的事情我们不知道, 但是我们知道他做的和什么有直接关系 . 没错, 有直接关系的就是从覆盖原式第一个 X 开始到 M 亮中间的这些字符 .

这一大堆字符里面只有很少的几个是用于引出效果的【控制字符】, 其他对效果类型无影响 (条形码最典型了, 三个控制出条形码, 剩下的拼图案)

大概最基础的效果有以下几类（这些在[视频](#)里面全都有，就不配图了）

刷出条形码

把字符翻译成 ASCII

爆机（横线之类）后面会举例

无限闪屏（肯定和光标的闪烁有关）

条形码刷屏

数据爆机（猜测是内部运算状态显示出来了）

移动应用堆栈的生长点（最重要的，乱点模式，异常模式，Es 系列可以升级到 991 的原因）

后面异常模式各种奇葩现象的基本都是这几类效果的组合

每一类效果都有固定的字符对应，称它为 [控制字符]

控制字符只有很少的几个。条形码的就是 log 根号 log，控制翻译 ASCII 的可以是第五十三个大于号，五十四小于等于号，也可以是之前说的 5 乘号乘号（还有很多未知的）。实际上 M 亮之后 AC 左等于，第一次按等于时读取控制字符的位置就是 53,54,55,56 这几个（有时只有前两个就管用了）

然后我会逐一解释上面列出来的各种效果

条形码的具体操作上一节讲过了。于是直接从内存的角度说。PLUS 的计算器啊，在输入超过限制的时候会把超出的一部分（就是临时储存区）循环的填充进后面的内存。So，条形码拼字会有循环现象。因为只有 100 个字符循环，但是显存总共有 372 个字节在控制。于是就循环了三次多。

然后翻译成 ASCII 的话，由于 PLUS 的模拟器没有办法异常或者溢出。具体的机制还不清楚

花屏爆机根据推测是芯片运算的状态显示到屏幕。（最简单的操作方法看视频的最快爆机）而最后爆的横线和电容放电有关，纯硬件现象。可以试试在最后爆横线的时候把电池拿出来，横线不会消失的。最简单的爆机就是在光标卡住之后全部按 sin(到 M 亮回去等于，爆横线。

无限闪屏：覆盖 X 那个开始，44 幂 44 根号 44 幂 44 根号到 M 亮

数据爆机 (视频有)

在 CALC 等于完光标卡住的地方先按四个数字 , 然后按根号根号 2, 根号根号 2 重复到 M 亮 , AC 左等于 , 左等于到屏幕花屏 , 等待几秒钟 . 数据爆机时间非常长 , 会有两轮 .. 最后会自动关机 .

然后重头戏 , AC 左等于的时候 , 按等于号的作用是【移动堆栈的生长点】

开机 , 然后最前面操作都一样到长按右键那里 , 这一次开始狂按根号按到 M 亮 (如果一直按下去最后会第二次出现最开始的原式 , 而且输入恢复正常) , 然后 - AC - 左 - 等于 - 左 - 等于 - 左 - 等于 第十五次按等于号后再按左会发现屏幕下方有乱点 . 这个时候按 shift-MODE-1-1 回到数学模式 (MathIO) , 刚才按了 15 下等于 , 现在是【乱点模式】 , 所以此模式的名字叫【15 乱点】 , 按多少下等于号就叫多少乱点 . 随便按几个左括号下面的乱点会上涨 (很好玩对吧喵 ~) . 按太多会死机的 . 死机就只能重新再来一遍了 . 然后举一个 15 乱点里异常模式的例子 ~ .

先到 15 乱点 , 十一个左括号 (乱点刚刚好淹到括号脚底下) , 然后按一个右括号 , 这时进入了一个新模式 , 光标上方有会变化的点跟着光标在闪 . 点本质上是二进制数字 , 光标每闪一下这个数字就 +1 , 这就是定时关机的原理 , so , 有这种现象的模式统称【计时模式】 , 光标有活动时就从头开始重新计数 . 当这个数字达到足够大的时候就触发关机 .

计时模式的很多性质都和时间的有关 , 而且计时模式基本都是比较活泼的模式 . 好吧先到这里 , 继续说 , 在这个计时模式下 , 输入 $10+10$, 然后按等号 , 右下角出现小方块 , 然后根号 - 等于 - 等于 - 根号 - 等于 . 然后 AC . 光标自己在跑 .. 嗯这个视频里面有的 . 叫自动复制模式 (视频有) 回到数学模式 , 按两个左括号 , 分数线 , 然后删除到什么都不剩 . 按乘号 , 会发现出来的 N , 对 , 这就是 ASCII 模式 (之一)

另一个 ASCII 模式的进入方法为 : 进入 16 乱点 , 10 个 Ans , 两个带框的 log , 在最下面的框里按正括号到换模式 . AC , 右 , 右 , 删掉小小的括号和积分号 . 也是按乘号为 N . (视频有 , 上面 LOVE YOU FOREVER 图就是这个模式里的

计时模式是【异常模式】的一种 , 异常模式有非常非常多 . 下面举一个最典型的异常模式的例子



在 15 乱点, 一个带框的 log, 然后在下面那个框里按左括号到乱点消失。

这个模式就是一个非常标准的异常模式。异常模式的标志是按一定数量的函数字符 (说明书有讲), 比如左括号, 分数线之类会出现乱码。推荐两个经典的异常模式: [第一个](#) [第二个](#)

那么, 你有没有想到, 按一定数量出现乱码和刚才乱点模式的乱点上涨的源头是相同的呢。So, 这两种现象的原因都是堆栈的数据生长点移动。

看到应用堆栈有一堆 Z (内存结构图接下页), 下面有一些数据, 这里的数据主要和输入的 [函数] 字符有关, 就是所有含有正括号的字符, 比如正括号, 分数线, sin(等等. 这个地方的数据是整个内存里面唯一往上涨的, 比如按几个正括号, 会发现这些数据从下往上覆盖了一部分 Z 如图. 一个正括号这里变化远远不止一个字符, 然后, 往上涨有一个初始的生长点. 姑且叫生长点吧. 生长点上方, 有一个固定的位置的 [限制], 这些数据不能越过 [限制] 往上继续生长. 而且这些数据是强制刷新的, 如果你在软件里面修改了他们, 和输入区不同这里的马上会恢复 ..

非常多的溢出各种现象都是这个生长点位置移动惹的.

[溢出] 的实质, 就是这个生长点跑到 [限制] 的上面去了, 于是 [限制] 就完全不起作用. 假如生长点的位置在 [限制] 以下但是比默认位置高, 那么就是 PLUS 的【降级模式】, 输入少量字符就顶到 [限制] 无法继续输入, 如果在 [限制] 以上显存以下, 就是【乱点模式】, 因为上涨的数据会覆盖到显存显示出来, 如果在显存以上输入区以下, 就是【异常模式】, 输入一定括号什么的, 这个生长点长出来的数据会覆盖到输入区, 于是按一定字符就会出乱码. 同时, 输入区下方是临时储存区和变量. 所以, 假如在异常模式按四个正括号出乱码, 在按三个以后 AC 左, 也一定会有乱码在里面., 而且变量全部都会数据异常 (变量储存后面有详细讲的)

模式划分, 应该按照这个生长点的位置, 越高那么模式的所谓 [溢出程度] 就越高

降级模式: 根号完左等于左等于的时候按十二次以下等于就回到 math

即时显存！

z/s

应用堆栈(异常的源头)

ZZZZZZZZZZ

[illegible]

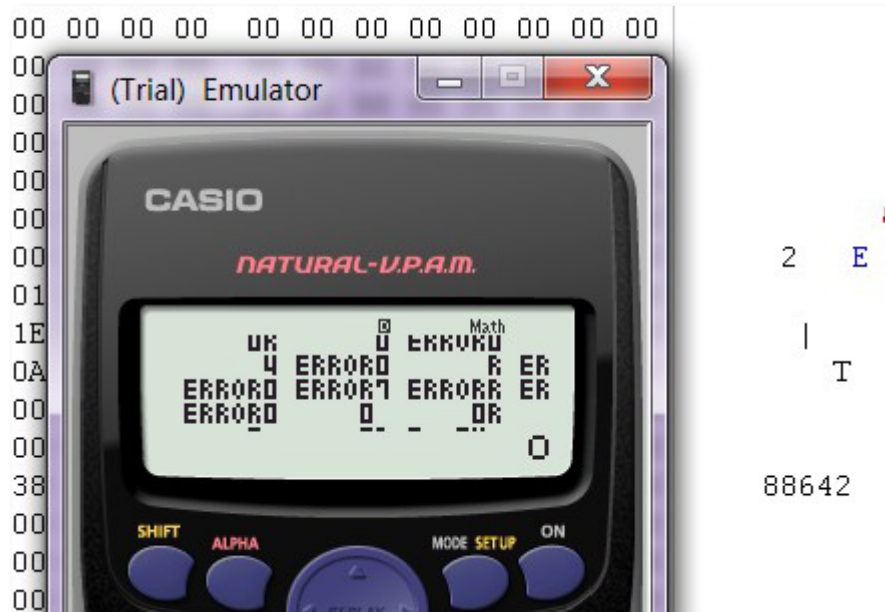
00	00
00	00
00	00
00	00
01	00
11	01
00	00
00	00
00	00

光标 | T

I

然后，大家都知道计算器的内存里肯定是有一部分用于控制模式，状态之类的。这一部分内存存在输入区上面（看图!!）。现在发现最有用的就是主模式控制字符和子模式。（什么字符代表什么模式是程序设定的..）. 比如一般

情况 COMP 模式下那个地方数据是 C1,ASCII 是一个有飘号的 A. 如果各位现在有模拟器和 WINHEX 之类的东西在手,可以试试把这个带飘号的 A 改成大写字母 E. 然后按 ON. 白屏,然后全部都是 ERROR. 如图 .E 实际上是 EQN 模式的主控制字符,但是当子模式定义异常的时候就会这样. 如果把 E 后面那个字节的 16 进制数据改成 01, 就是正常的 EQN 界面 ..



时候就会这样. 如果把 E 后面那个字节的 16 进制数据改成 01, 就是正常的 EQN 界面 ..

改成 E 以后的情况很好玩, 那么在实体机上怎么实现呢? 实体机是不能直接访问内存进行修改的, 于是就要利用一些东西. 就是那个 r

具体操作

991+ 先到第两百个不稳定字符那里, 打出 cr(STAT 模式下的小 c 和 r, 不稳定字符填入根号很快就到了), 然后删掉 cr 前面所有的只留下这两个, 然后按 [=], [右], [del] 删掉 c, 然后 5 个 [(], 再 [1], [(], [1], [(], [1], [)], [M+], 最后按 on, 白屏. 过大约一分钟屏幕会全部都是 ERROR

p.s. 白屏的时候可以告诉别人计算器坏了然后低价收来什么的。

然后, 那个主模式控制字符填入很多别的东西, 比如主控制字符填入 ASCII 感叹号就是【68 模式】。【68 模式】按 ON' 不会清除向上箭头里面的东西 (算完以后按 ON 上箭头还在) 而且 68 一般是线性的, shift-9-1- 等于 -AC 清除 SETUP 后向上箭头里的算式不会被清除, 可以在 MATH 下继续调出线性算式使用

fx-82ES PLUS 的异常

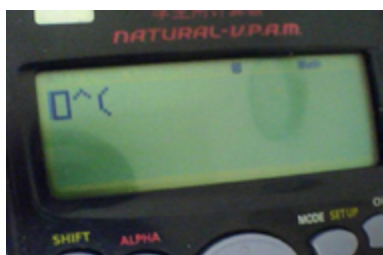
曾经 ES 系列是从 82 这个版本发家的。PLUS 是 991, 那么 82PLUS 有没有什么玩的呢. 答案是肯定的. 我们已经知道 PLUS 系列的异常必须通过输入超过限定的字符然后按等于号移动堆栈进入异常 / 乱点模式。搞出 82+ 异常这曾经是贴吧所有同学的梦想 = =

要输入超过限定的字符必须找到入口。991+ 是那个 CALC 的异常错误，但是 82 的就稍微诡异一点了。

计算器里面有一些基础字符有很怪异的性质，r 是一个，还有框。嗯对就是分数线上下都有那个。独立的框可以像 991+ 前面操作那里一样卡住光标。卡住光标就意味着可以输入超过限定数量的字符了。

搞出框的方法有多种，[掰电池法](#)，[干簧管短接电容法](#)，还有 [68 法](#)。这里介绍 68 法，因为它完全不涉及硬件，而且已经用于进入异常了。

开机 -Shift-9-3 (resetALL) 然后等于和 ON 一起按，然后重复此步骤...直到右下角有一个 0 出现，随便算一个东西 ON 以后向上箭头不消失 .. 恭喜你进入了 68 模式 .. 由于计算器芯片的速度不是很快...在恢复默认的过程中强行断电终止导致设置不完全 .. 就有可能进入 68 模式了 (很朴素的思想对吧 ~). 当然，这个比较靠【人品】。【人品】好的时候也许按几次就进去了，不好的话...本叶绿素同学曾经二十分钟都....T_T，不过相信一定可以进去的就对啦 ~



进入 68 以后，按 [1]-[幂]-[1]-[=](就是计算一的一次方). 然后 [shift]-[9]-[1]-[=]-[AC](回到 MathIO 模式了，上箭头还在)。按上键，会显示刚才算的 1^1 (这个式子，[右]-[DEL](在 ^ 前面出现了一个框) [左] (光标在最后面) [DEL][DEL] (就是删掉其他只剩下框) [左] (光标在框前面) 然后 [长按右键] !

光标被卡死，输入任何字符光标不会再动了。如果这时候全部按 9 到 9 重新从屏幕正常显示，AC 左等于可以到自检界面

还是从长按右键卡住光标那里开始。按三十二下 [分数线]，然后按 [8]-[8]-[8]-[shift]-[Ans]-[3]-[8]-[8]-[8]-[shift]-[Ans]-[3] (就是 888g) 重复此步骤到 M 亮。然后 AC 左，最



前面四个应该是 8g88. (如果不是请从框那里重新执行以上步骤)。是的话，那么按 15 下等于号，AC。这时候是 15 乱点，这里和 991+ 有许多性质相同的地方。

82PLUS 唯一的升级 BASE-N 模式 (视频有)

15 乱点，十一个左括号一个右括号，AC 即进入计

时模式(和 991+ 一样的)。然后迅速的按[幂]-[分数线]-[等于]-[AC]就进入了 BASE-N(进制转换)模式。这时候按 shift 9 2 = AC 清除 MEMORY 就可以正常使用功能了(四个进制状态,16 进制的 A-F 都可以正常输入使用)。但是如果 shift , 3 进入 BASEN 的菜单,会发现全是乱码。能进 BASE-N 的实质,其实也是按等于号导致模式控制字符(索引)被直接修改。这种按等于号以后的运算真的比较诡异而且近乎无法研究...

82+ 可以进入 BASE-N 也就是部分升级,那么有没有可能完全升级到 991+ ? 这曾经是一



个争论过很久的问题.. 正确答案是不能. 原因是程序不一样

首先是自检. 在 SHIFT 和 7 一起按住的情况下按 ON, 屏幕即会显示 DIAGNOSTIC Press AC. 这时按 9 就进入自检了. 按几次 shift 屏幕会出现版本信息. GY-450 VerE 中, G 应该是代表 global(全球), 450 是 82+, 453 是 95+, 455 是 991+, 还有一个是 350+ 好像是 452. 国行四个版本. 下面一行 SUM 总结? 具体不明, 但是 82 和 350 的 SUM 很接近, 用于异常的方法也一样. 也许说明一些问题. 现在可以用来玩溢出之类的 PLUS 系列只有 E 版 82+ 和 E 版 991+. 如果你的计算器自检是 VerB 或者 VerA 的假货, 玩这些就不要想啦.

然后从内存方面, ES 系列的模拟器搜索菜单字符串是可以搜到四个版本的菜单的, 82, 500, 991, 还有一个是六模式但是没有发行. 但是, PLUS 系列的模拟器一个版本只能搜索到一个菜单.

运算功能方面, 你已经知道了 82+ 如何进入异常, 那么可以用异常模式的乱码调出一些本不属于 82+ 的字符进行运算测试, 会发现几乎没有能正常用的. 尤其科学常数貌似都是用来测试功能极限的数值.

也许你会问，那为什么 BASE-N 可以用那些功能呢？这一点从 CASIO 官方客服得到了答案，同时也证明了程序不一样的猜想。而且，模拟器搜索版本信息不同版本的都不一样
不同版本模拟器搜索 GY- 字符串 的结果 官方全版本 PLUS 模拟器下载 (ISO)

95SG GY-458B

82AU GY-459B

86DE GY-462C

92A GY-468A

92B GY-461B

500VN GY-460C

82/85/350 GY-450B

from us.

If you are talking about FX-82ES PLUS and FX-991ES PLUS,

both models' program is different.

However,BASE-N MODE is basic function.

If you are interested in our FX-82ES PLUS and FX-991ES PLUS,please access following site in Asian area.

<http://www.casio-intl.com/calc/scientific/ntd/>

Basic function。基础功能，说明 PLUS 的程序非常有可能是在一个基础程序上面分版本开发的，而 BASE-N 的部分属于基础程序部分，字符库，菜单字符也属于这基础部分，运算程序不属于。菜单乱码说明菜单也应该是后面才定向的。

82+ 的 ASCII 模式

1. 进入 36 乱点
2. 任意的输入一个数字，这里我们用 8
3. 打出 $8 \rightarrow C \rightarrow X$
4. 删去 8
5. 在 X-1 无限输入 “ (” 溢出代码
6. 在溢出的代码中找到 $\rightarrow r \angle \theta$ ，然后删掉其他代码，剩下 $\rightarrow C \rightarrow X \rightarrow r \angle \theta$
7. 在 $3 \sqrt{}$ 中无限输入 “ (” 溢出代码
8. 在溢出的代码中找到 $\rightarrow a+bi$ ，然后删掉其他代码，剩下 $\rightarrow C \rightarrow X \rightarrow r \angle \theta \rightarrow a+bi$
9. 接着在 $\rightarrow C \rightarrow X$ 中间打 \div (或者打 0)，在 $\rightarrow r \angle \theta$ 后面打出 $EY \div$
10. 在 $\rightarrow C$ 前面打出 100^2
11. = =

直接复制的原帖。<http://tieba.baidu.com/p/1893034938>。我自己没怎么研究，但是验证过确实可用

变量储存 (灰常重要的独立部分)。

都知道计算器有变量的功能, ABCDEFXYM 对吧 ~ 但是他们是怎么存的呢.... 如何利用这个东西弄一些奇葩的玩意 .. 继续

这一部分感谢 @ 喵喵叫的绵羊 .. 讲的很清楚了 .. [原帖](#)

78180	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
78190	00 00 00 00 00 00 00 00	58 3D 69 58 2C 31 2C 31 X=iX, 1, 1
781A0	74 39 00 00 00 00 00 00	00 00 00 00 00 00 00 00	t9.....
781B0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
781C0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
781D0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
781E0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
781F0	00 00 00 00 00 00 00 00	00 00 00 00 58 3D 69 58	Y=iX
78200	2C 31 2C 31 74 39 00 00	00 00 00 00 00 00 00 00	, 1, 1t9.....
78210	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
78220	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
78230	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
78240	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
78250	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
78260	00 00 00 00 00 00 00 00	7A 00 00 00 00 00 00 00	2.....
78270	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	Ans的10个字
78280	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	节
78290	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
782A0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
782B0	00 00 00 00 00 00 00 00	00 00 01 71 66 70 00 00	

21 A2 80 00 00 00 00 00 04 01

① ② ③

① 这半个字节（记作 a）决定了默认的显示方式。

0:标准

2:分数显示，数据中若有 A 即为分数线

4:度分秒显示...

8:根式

6.7.9.A.B.C.D.E.F:ERROR，然后这个字节会变成 F3，后面都是 0

其他目前不明

② 这七个半字节没半个字节前面写一个 3，对应 ASCII 中的哪个，就显示哪个。

如：26 34，即对应 32 36 33 34，即 2、6、3、4 这 4 个数字。（所以可以理解为他们就是 10 进制的！）如果出现了 ABCDEF，就会在结果中显示 3A 3B 3C...他们对应的那几个符号。。其中 A 可以作为分数线。（这个可以称之为短码。。）

1) a=0 时，对应 15 位有效数字

2) a=2 时，里面的 A 前分子，A 后分母。如果 A 有两个的话……目测是带分数...

3) a=4 时，对应与其相等的小数的 15 位有效数字

4) a=8 时，（十分复杂，尚未完全解出）末字节是分母，往前一个是根式的系数，往前两个字节是根号下的东西（后三位有效，前面的那位不知道什么东西）再前一个字节好像和分母有啥关系没搞出来，再前一个是另一个根式的系数或有理项，再前一个以及 a 后面的半个字节这三位是另一个根号下的东西。如果是 0 01，后面的字节就是有理项（废话。。）

③ 这两个字节：

1) a=0,2,4 时，前字节决定小数点的位置（所以也决定了科学技术法的显示），后字节决定符号（01 正，06 负，其它的值貌似也有用但是没发现。。）利用这两个字节可以使分子或分母上使用科学计数法..

2) a=8 时，前字节决定后一个根式的符号（01 正 06 负），后字节决定前一个根式（或有理项）的符号（也是 01 正 06 负）

现在你应该明白第一步的 二分之一等于 是什么作用了

延伸（发挥你想象力的时候到了...）：

可以弄出来的字符如下：

字符	Hex	短码	字符	Hex	短码
0	30	0	8	38	8
1	31	1	9	39	9
2	32	2	:	3A	A
3	33	3	;	3B	B
4	34	4	<	3C	C
5	35	5	=	3D	D
6	36	6	>	3E	E
7	37	7	?	3F	F

【有奇怪符号的结果可以正常地存进各个变量中】

但是，由于这个区域位于不稳定字符的后方，所以你不能使用不稳定字符打出其他字符，只能使用可以直接打出来的字符（比如 21 对应的是 Math 中的那个奇怪的方框，各种郁闷不能用，最后我放弃了零分之一的想法而试着打了零分之二……）

建议先用模拟器试试需要用的字符能不能打得出来，顺便也免得查表了。。

So. 假如制造一个违反规定的变量会怎样 :991+

前面都一样, 狂输数字至 M 刚好亮, [DEL] (这时 M 灭), [sin()], [AC], [根号], [RCL], [M], [SHIFT], [RCL], [M], [ON], 此时 M 即“死机变量” (991 也可直接在不稳定字符后 M 的存储位置填入 "mp1234567sin(", 则 M 即为死机变量)

性质: 此变量的数据是允许储存的, 但其指数部分的显示却是不被允许的 (因为变量的第九个字节 (控制科学计数法的指数) 是大于 99 的, 且第十个字节为空 (即 00, 控制数字及指数正负), 详见 @ 喵喵叫的绵羊 关于这方面的帖子), 因此, 当此变量被要求包括指数部分的显示时, 就会死机 (例如, 正常模式下, RCL M, 死机)

进入 EQN 模式, 随便在哪里填入 M 后, 按 ON。 (by 946994919)

永久白屏 (相当于 119)。解除方法太过简单这里就不说了。

最后一点点说一下键盘扫描

键盘没按下去之前是断路, 按下之后该条电路接通. 电流从固定方向流入, 从另一端流出. 电路板上有很多银色的圆点标着 KI (KeyInput) 或者 KO (KeyOutput)。根据这个判断按下的是哪个键. 四按一松出 cos(., 0, 减号, 等于号在下表上刚好围成一个方框, 1 可以理解成用于占位. 能够替代按出的只有框左下角那个. 其他所有按键都一样. 只要四个按键在表上围城一个方框, 就可以四按一松出来 (内容到此结束, 希望 .. 不会想睡觉吧 ..)

fx-es PLUS系列键盘按键I/O关系

KI \ KO	1	2	3	4	5	6	7
1	1	2	3	+	-		=
2	4	5	6	x	÷		Ans
3	7	8	9	DEL	AC		$\times 10^x$
4	RCL	ENG	()	S \leftrightarrow D	M+	.
5	负	...	hyp	sin	cos	tan	0
6	分	$\sqrt{\quad}$	x^2	x^\square	log	ln	
7	CALC	$\int \square$	\leftarrow	\downarrow	x^{-1}	$\log \square$	
8	SHIFT	ALPHA	\uparrow	\rightarrow	MODE		

WRITTEN BY 2422

y/s

**终于弄完了啊!!!
排版排到想死啊!!!
一天一两小时搞了快半个月啊!!!
封背景配色调无数次还各种违和只能这么凑合了啊!!!
明明感觉没多少东西写出来就 1W 多字了啊!!!**

**就不点 PLUS 研究中的各位大神的名了 . 装不下 ...
去看 PLUS 精品贴列表的发帖人部分就知道了的说**

**PLUS 这段时间年我们在贴吧玩得很开心 . 结交了很多各种各样的朋友
甚至发现一起的某 NB 吧友中居然有现实生活中隔壁班的同学
成果和技术永远没有友情重要。**

嗯就这样吧

全篇完

by yls

