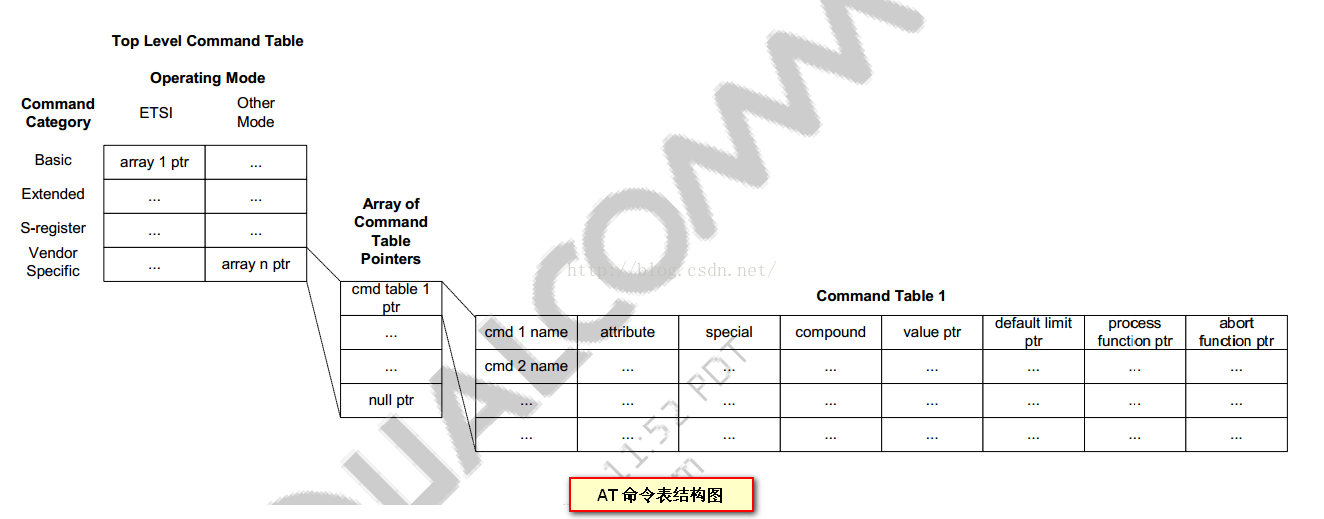
## AT命令表（ATCommand Table）:



AT命令的处理是由命令表驱动的，ATCOP实现的命令表是一个分级的表结构，主要分为：

·主表（master table）

·子表（sub table）

·命令表（command table）。

1、其中主表是一个二维的数组，数组的行表示AT命令的分类，分为：

·基本AT命令（basic\_table）

·寄存器AT命令（sreg\_table）

·扩展AT命令（extended\_table）

·厂商AT命令（vendor\_table）四大类；

1. 数组的列表示是ETSI模式还是其它模式的AT命令。
2. 命令表表示具体的命令（如果扩展AT 命令需要修改此表）

## ·AT命令表分类具体介绍：

### 1．基本命令表

基本命令的格式为：<command>[<number>]

其中<command>或者是单个字母（A-Z），或者是“&”字符接单个字母。

<number>是一个十进制数，可以是一位，也可以是多位。<number>最前面的0会被忽略。默认<number>为0。如果一个不带<number>的基本命令带了<number>，则返回TOO MANYPARAMETERS。

**比如：”&C” , “E”,**

### 2．寄存器命令表

所有以字母“S”开头的命令统称为S寄存器命令，格式如下：

S<parameter number>? S<parameternumber>=<value>

S 寄存器命令名由字母“S”接上一个十进制数构成，这个十进制数称为寄存器序号（parameternumber）。如果寄存器序号不被识别，说明不存在该命令，返回COMMANDNOT SUPPORT。

每个 S寄存器保存一个字符。命令名后面如果接“?”表示是READ命令，返回此S寄存器当前保存的字符的ASCII 码值，以3 位十进制数表示，位数不足的前面补0；如果接“=”表示是SET命令，将<value>值对应的字符替换此S寄存器保存的字符。

比如：**”S0”，”S2**”

### 3．扩展命令表和厂商提供的命令表

扩展命令均由“+”开头，厂商定义的命令也是由一个特殊符号开头，例如“$”，“%”等。本文中所实现的命令均为扩展命令。所有的扩展命令和厂商定义命令又可以分为两类：Action command和Parameter command。

#### 1）Action command

Action command 指完成某个具体的动作，而不仅仅是与MS 本地的参数打交道的命令，例如AT+CMGS 等。Action command 可以带参数也可以不带。Action command 包含EXECUTION 命令和TEST 命令。

**（1）EXECUTION命令**

EXECUTION 命令格式如下：

不带参数：<name>

带 1个参数：<name>[=<value>]

带多个参数：<name>[=<compound\_value>]

<compound\_value>表示多个参数，中间以“,”分隔。对于有默认值的参数，可以在命令中省略，此时以默认值代替。

如果所有的参数都省略，则<name>后面的“=”也一并略去。如果<name>不被识别，则表示此命令不存在，返回COMMAND NOTSUPPORT。<name>可识别的前提下，如果不能带参数的命令带了参数，或者参数个数超出定义，则返回TOOMANY PARAMETERS。

**（2）TEST命令**

TEST 命令格式：<name>=?

如果 MS不能识别<name>，返回COMMAND NOT SUPPORT。如果MS可以识别<name>，且命令是不带参数的，则返回OK。如果命令带参数，则先返回各个参数的可取值范围，最后再加上OK。

#### 2）Parameter command

parameter command包括与MS本地的参数打交道的命令，这些参数有些会影响到Atcion command的执行。Parameter command又分为SET命令、READ命令和TEST命令。

**（1）SET命令**

命令格式为：带1个参数：<name>[=<value>]

带多个参数：<name>[=<compound\_value>]

SET命令用于设置参数。<compound\_value>表示多个参数，中间以“,”分隔。对于有默认值的参数，可以在命令中省略，此时以默认值代替。如果所有的参数都省略，则<name>后面的“=”也一并略去。如果<name>不被识别，则表示此命令不存在，返回COMMAND NOTSUPPORT。<name>可识别的前提下，如果不能带参数的命令 带了 参数 ，或者 参数 个数 超出 定义， 则返回TOO MANYPARAMETERS。

**（2）READ命令**

命令格式：<name>?

READ 命令用于读取参数当前值。

**（3）TEST命令**

命令格式：<name>=?

如果 MS不能识别<name>，返回COMMAND NOTSUPPORT。如果MS可以识别<name>，且命令是不带参数的，则返回OK。如果命令带参数，则先返回各个参数的可取值范围，最后再加上OK。

## 命令表定义的各项属性

/\*---------------------------------------------------------------------------

The following data type is the data structure for a command table entry.

name - The name of the AT cmd, including the leading "+", "$"

or "$" and the terminating NULL.

attrib - A 32-bit mask that specifies one or more command

attributes.

special - The "special processing code," if any, or SPECIAL\_NONE.

The special processing codes are defined in this file.

A unique special processing code must be assigned to

each AT cmd that requires some special processing.

compound - The number of arguments, for numeric commands, or the

maximum string length (not including the terminating NULL),

for string commands.

val\_ptr - Val pointers moved into strcuture and mapped via [get](http://opengrok.tcl-mobile.com/Athena_p/s?path=get/)/[set](http://opengrok.tcl-mobile.com/Athena_p/s?path=get/set) API.

def\_lim\_ptr - Defines the default value and the allowed range for the

arguments.

proc\_func - Pointer to function to be called to process the command.

abort\_func - Pointer to function to be called to abort the command.

If the pointer is NULL, command is not abortable.

---------------------------------------------------------------------------\*/

typedef struct dsati\_cmd\_struct

{

char \*name; // AT cmd 的名字，包含"+", "$" 等

uint32 attrib; // AT cmd 的属性

byte special; // AT cmd 的 special processing code

byte compound; // 传递的参数个数(若参数为字符串，则是它的最大长度)

uint16 cmd\_id;

const void \*def\_lim\_ptr; // 定义了参数的默认值以及取值范围

} dsati\_cmd\_type;

## AT Command的测试过程

以后再补充