



# SIM800系列\_Embedded AT \_编译环境说明

GPRS 模组

芯讯通无线科技(上海)有限公司  
上海市长宁区金钟路633号晨讯科技大楼B座6楼  
电话: 86-21-31575100  
技术支持邮箱: support@simcom.com  
官网: www.simcom.com

文档名称:	SIM800 系列_Embedded AT_编译环境说明
版本:	1.03
日期:	2020.6.15
状态:	已归档

## 版权声明

本手册包含芯讯通无线科技（上海）有限公司（简称：芯讯通）的技术信息。除非经芯讯通书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播，违反者将被追究法律责任。对技术信息涉及的专利、实用新型或者外观设计等知识产权，芯讯通保留一切权利。芯讯通有权在不通知的情况下随时更新本手册的具体内容。

本手册版权属于芯讯通，任何人未经我公司书面同意进行复制、引用或者修改本手册都将承担法律责任。

### 芯讯通无线科技(上海)有限公司

上海市长宁区金钟路 633 号晨讯科技大楼 B 座 6 楼

电话：86-21-31575100

邮箱：simcom@simcom.com

官网：www.simcom.com

了解更多资料，请点击以下链接：

<http://cn.simcom.com/download/list-230-cn.html>

技术支持，请点击以下链接：

<http://cn.simcom.com/ask/index-cn.html> 或发送邮件至 [support@simcom.com](mailto:support@simcom.com)

版权所有 © 芯讯通无线科技(上海)有限公司 2020，保留一切权利。

# 关于文档

## 版本历史

版本	日期	作者	备注
1.00	2012-10-20	毛斌	第一版
1.01	2013-07-11	毛斌	适用范围更新，修改编译方式
1.02	2015-08-10	毛斌	适用范围更新
1.03	2020-06-15	方凡	修改文档结构和风格

## 适用范围

本文档适用于 SIM800 系列 Embedded AT 模块，包括 SIM800W, SIM840W, SIM800V, SIM800H, SIM800, SIM800M64, SIM808, SIM800C 的 Embedded AT 模块。

本文档描述了 Embedded AT 的编译环境及相关注意事项。

# 目录

版权声明 .....	2
关于文档 .....	3
版本历史 .....	3
适用范围 .....	3
目录.....	4
1 工具及代码结构 .....	5
2 编译 .....	6
2.1 安装工具 .....	6
2.2 编译流程介绍 .....	7
2.3 客户配置 .....	7
2.4 添加文件 .....	8
3 下载 .....	10
4 调试 .....	11
4.1 设置端口模式为 EAT_UART_DEBUG_MODE_UART.....	11
4.2 catcher 工具使用说明.....	11

## 1 工具及代码结构









编译器：RVCT，版本为：ARM/Thumb C/C++ Compiler, RVCT3.1 [Build 569]

编译环境工具：winmake，在 build 目录下提供

下载工具：SIMCom\_Multiupdate\_tool\_for\_MTK\_platform

下载工具：coolwatcher

目录结构：

 [build]	编译配置及 Makefile 脚本文件
 [core]	SIMCom 提供的头文件及库文件
 [demo]	Demo 源程序
 [Doc]	文档
 [src]	客户的源文件
 SIM800V128 bat	编译脚本 在命令行模式下运行脚本进行编译
 SIM800W64 bat	
 SIM840W64 bat	

## 2 编译

### 2.1 安装工具

RVCT 编译器安装成功后在命令行模式下输入 armcc 有信息返回：

```
f:\Module\SIM800_52D\MT6252MOD\eat\app>armcc
ARM/Thumb C/C++ Compiler, RVCT3.1 [Build 569]

Usage:      armcc [options] file1 file2 ... fileN
Main options:
```

使用脚本编译：

```
f:\Module\SIM800_52D\MT6252MOD\eat\app>build_SIM800W64.bat
```

注意：

build 目录下的编译脚本，user.mak，Makefile，option.mak，app\_build.mak 只能为 ANSI 格式，不能为 UTF-8 或者 unicode 格式。

如果在编译时出现类似错误：

```
user.mak:1: warning: NUL character seen; rest of line ignored
user.mak:1: *** missing separator.  Stop.
```

请修改编译脚本的编码格式为 ANSI。

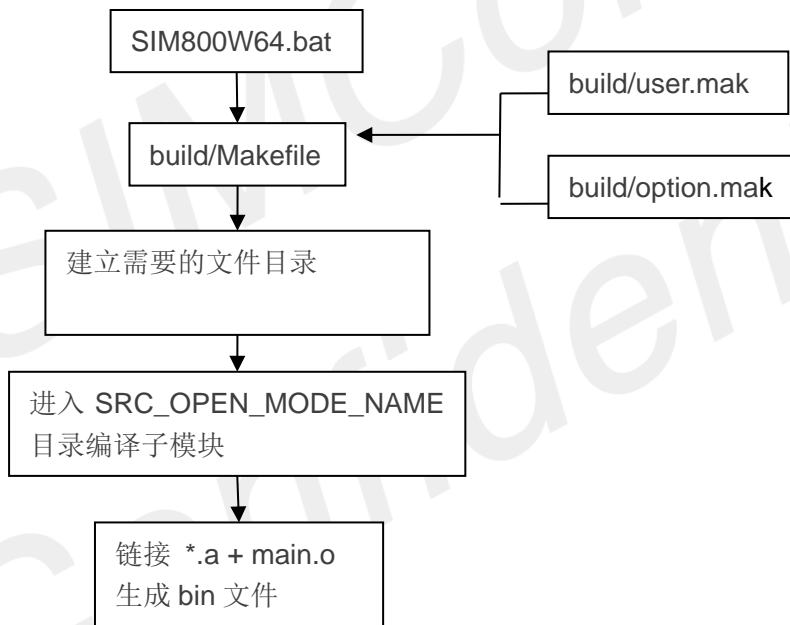
文件名(N):	user.mak
保存类型(T):	所有文件
编码(E):	ANSI

## 2.2 编译流程介绍

build 目录下文件说明:

文件名	功能说明	是否需要用户配置
user.mak	用户配置文件, 配置编译器路径, 源文件目录名, 编译模块名等	是
Makefile	系统入口 Makefile	否
option.mak	用于配置编译环境, 系统使用	否
app_build.mak	用于编译源代码, 系统使用	否

编译过程如下图:



生成的文件在 OUTPUT\_BIN\_DIR (user.mak 中定义) 指定的目录下, 文件名为 BIN (user.mak 中定义)。

SIM800H 生成的 app 文件在 core/SIM800Hxxx 目录下, 下载的时候重新加载 core/SIM800Hxxx/\*.cfg 文件, 再选择对应的要下载文件。

## 2.3 客户配置

客户一般需要修改 user.mak 文件中以下内容:

### #RVCT 3.1 Root Directory

DIR\_ARM = C:\Progra~1\ARM\RVCT

#源文件所在目录

SRC\_DIR = demo

#编译的模块名

SRC\_OPEN\_MODE\_NAME = adc

## 2.4 添加文件

客户添加源文件，例如，添加一个短消息处理测试程序，源文件有 main.c, app\_sms.c, app\_sms.h。

说明：**app** 源代码必须有一个 **main.c** 程序，用于 **app** 的入口。

注意：其他源文件名称不能有 **xxxmain.c** 命名。

步骤如下：

在 src 目录下新建 sms 文件夹。

- 文件夹包含 main.c, app\_sms.c, app\_sms.h。
- 从 demo/adc 中拷贝 Makefile 到 sms 文件夹中。

Makefile 中组织源码有两种方式，可以任选一种：

- a. 把所有“.c”文件全部编译

#### #First mothod#

S1 := \$(wildcard \*.c)

SOURCE:=\$(S1)

- b. 指定需要编译的源代码文件

#### #Second mothod#

SOURCE := main.c \

app\_sms.c

- 修改 user.mak 文件。

#源文件所在目录

SRC\_DIR = src

#编译的模块名



## **SRC\_OPEN\_MODE\_NAME = sms**

- 运行 sim800W64.bat all 编译 或者 sim800W64.bat clean 清除编译生成文件。

SIMCom  
Confidential

## 3 下载

目标文件有两部分，一个是 SIMCom 提供的 CORE 程序，第二个是客户自己编译的 APP 程序。正常情况下，出厂时模块中已经存在 CORE 程序，客户开发过程中只需要下载自己编译的 APP 程序。

注意：

### ※ 特别注意

- 如果 SIMCom 提供新的 CORE 文件给客户时，则需要更新 CORE 程序；并且 APP 也必须使用新提供的库文件重新编译,然后下载新编译的 APP 程序。
- 不同需求，可能使用不同的下载工具。

## 4 调试

有如下三种方法查看 log 信息:

- 客户自己实现接口，从通过 `eat_uart_write` 往串口输出 log 信息。
- 设置 **DEBUG** 端口模式为 **EAT\_UART\_DEBUG\_MODE\_UART** 模式，使用 `eat_trace()` 输出 log 信息，使用普通串口工具显示。
- 设置 **DEBUG** 端口模式为 **EAT\_UART\_DEBUG\_MODE\_TRACE** 模式，使用 **SIMCOM** 提供的 **Catcher** 工具查看 eat log 信息。

在 app 中使用 `eat_trace()` 接口打印 log 信息。catcher 工具使用说明见下文所述。

### 4.1 设置端口模式为 EAT\_UART\_DEBUG\_MODE\_UART

在 `app_func_ext1()` 接口中:

```
EatUartConfig_st cfg =
```

```
{  
    EAT_UART_BAUD_115200,  
    EAT_UART_DATA_BITS_8,  
    EAT_UART_STOP_BITS_1,  
    EAT_UART_PARITY_NONE  
};
```

```
eat_uart_set_debug(EAT_UART_2); // 使用 EAT_UART_2 作为 debug 口
```

```
eat_uart_set_debug_config(EAT_UART_DEBUG_MODE_UART, &cfg); // 设置 DEBUG 口为普通 UART 模式。  
默认设置为 TRACE 模式，输出 log 需要使用 Catcher 工具解析。
```

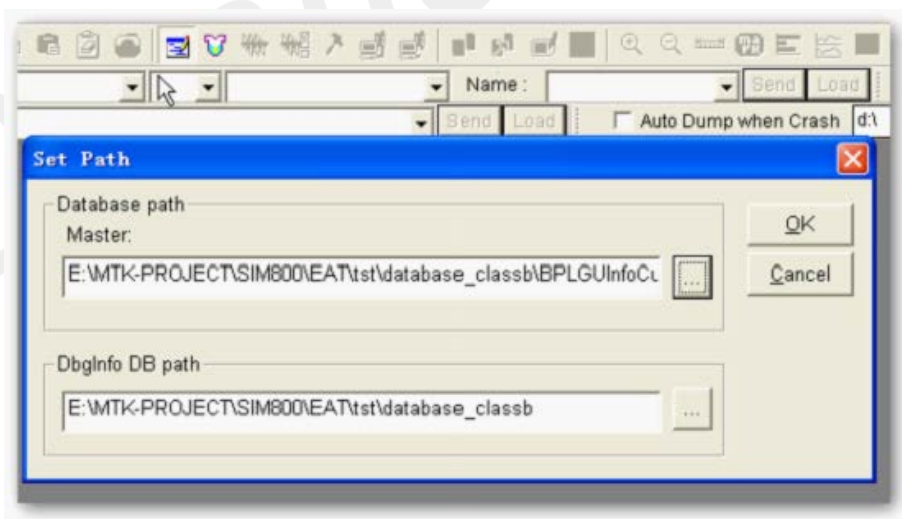
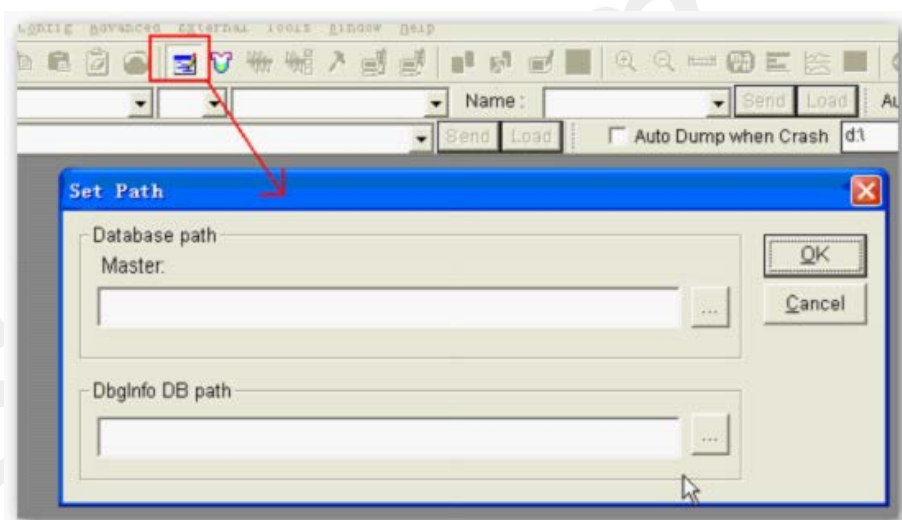
### 4.2 catcher 工具使用说明

- 这里使用模块的 **UART2** 端口抓 log，所以在 app 程序中需要如下设置:

```
void app_func_ext1(void*data)
{
    eat_uart_set_debug(EAT_UART_2);
    eat_uart_set_debug_config(EAT_UART_DEBUG_MODE_TRACE,NULL);
    .....
}
```

- 连接 UART2 到计算机,打开 **Catcher**。
- 选择数据库文件:

数据库文件由 SIMCom 提供, 类似 BPLGUInfoCustomAppSrcP\_xxx\_xxx\_EMBEDDEDAT

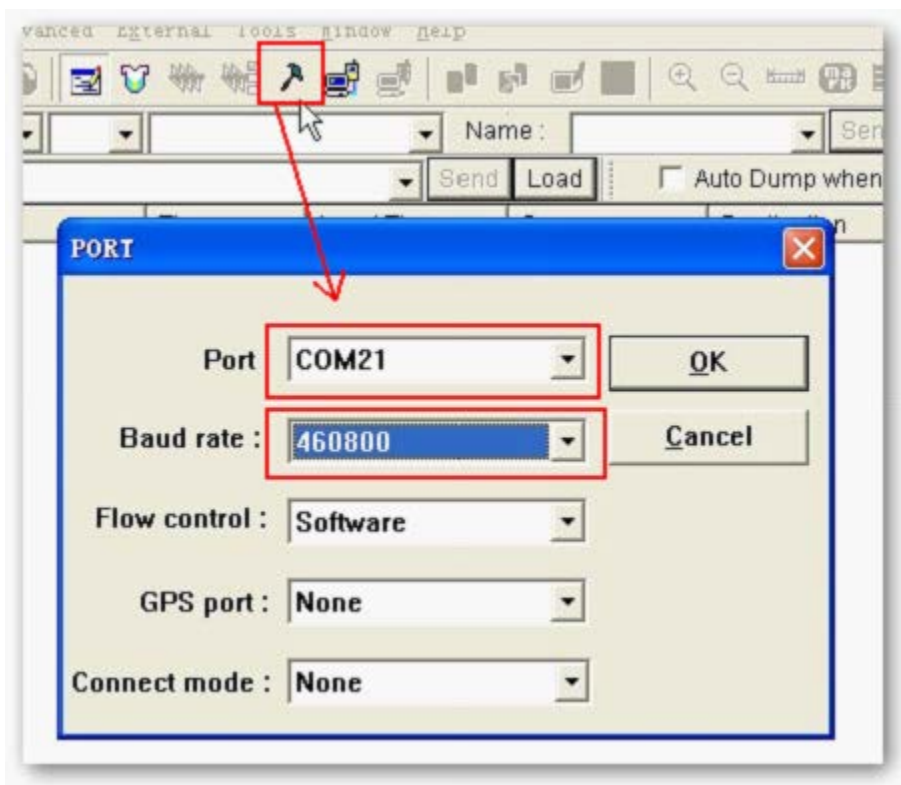


- 设置端口参数

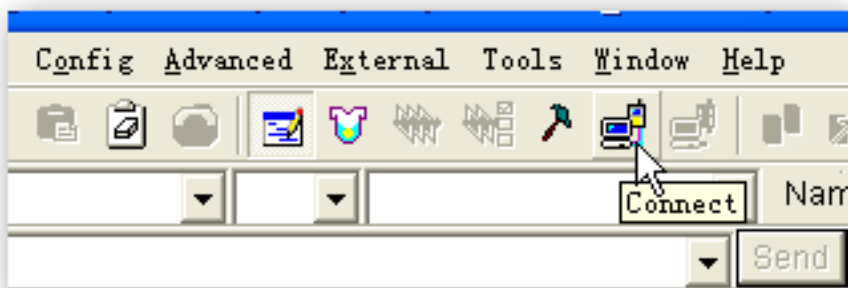
“Port”选择计算机上对应的端口, “Baud rate”设置为 460800。

使用普通串口线不支持 460800 波特率。

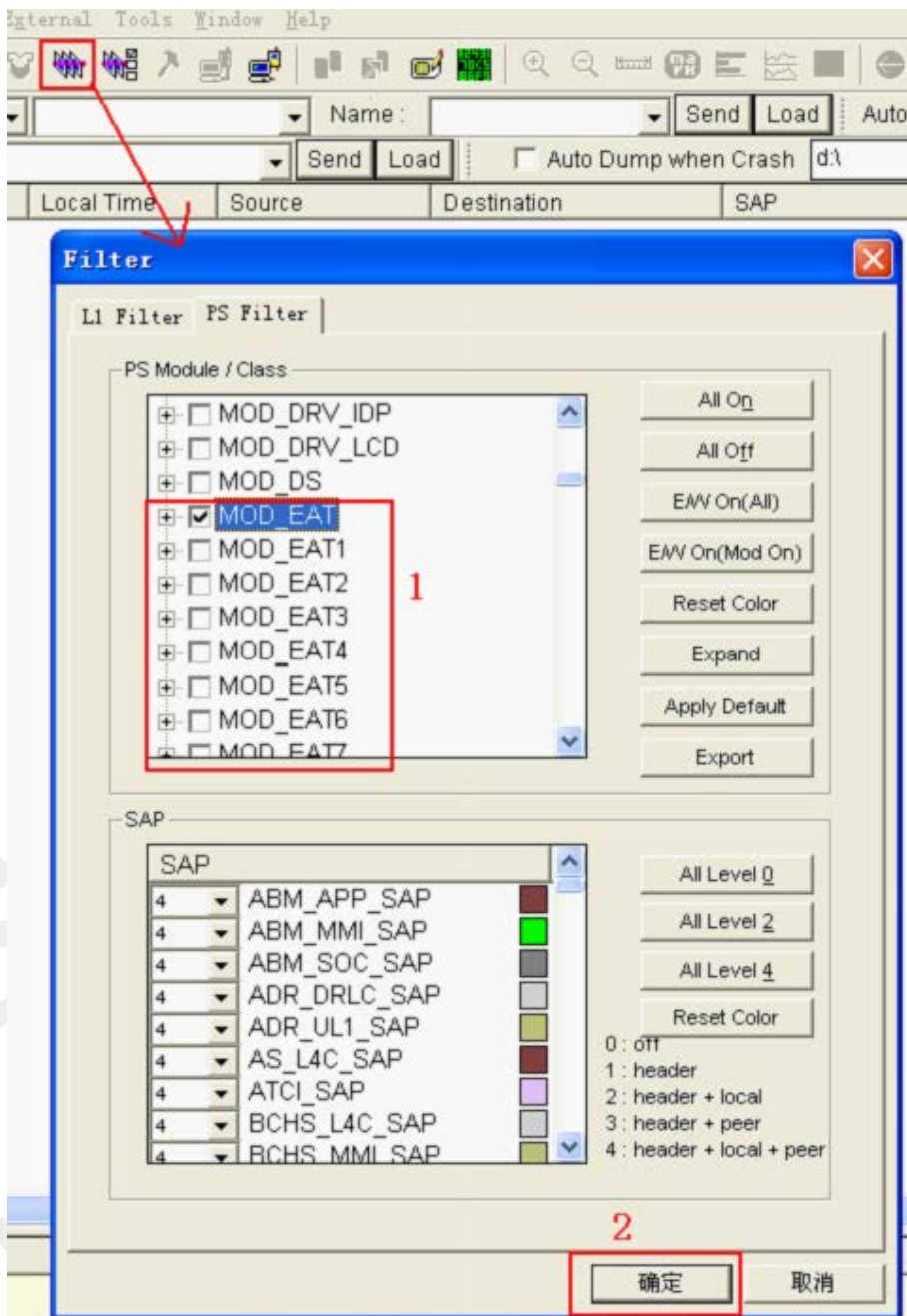
需要使用质量保证的 USB 转串口线, 否则可能因为波特率误差而连接不上。



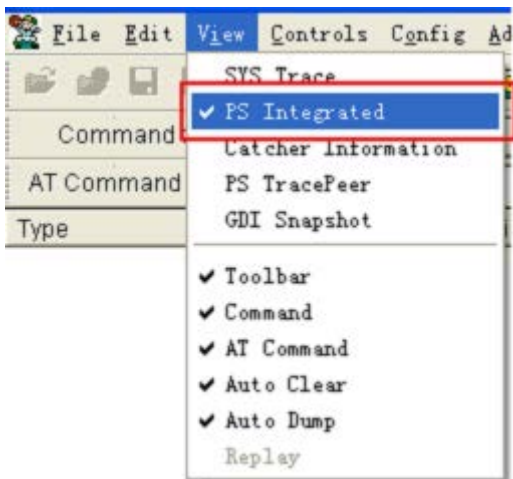
- 点击“Connect”开始连接



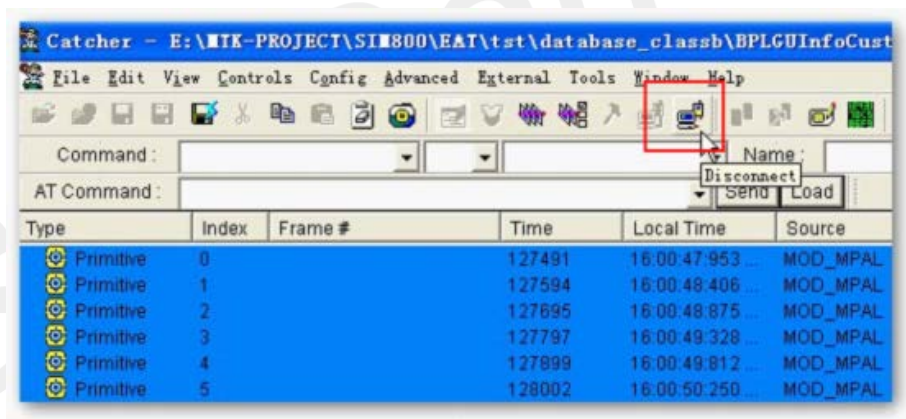
- 设置过滤器



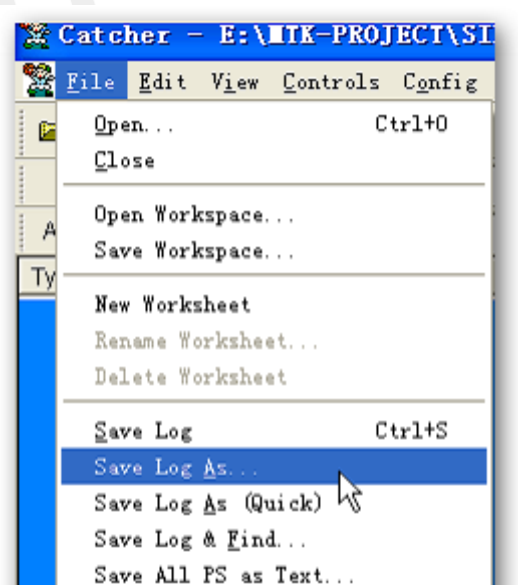
- 开机抓 log，在 View 中选择上 PS Integrated，在主窗口会看到 log 内容。



- 如果需要保存 log 则需要先断开连接，点击“Disconnect”按钮。



- 通过“File”菜单中“Save log as...”按钮保存 log。



- log 信息如下图:

Primitive	298	1636	15:33:15.360 ...	MOD_DRV_HISR	MOD_EAT	DRIVER_P8_SAP	MSG_ID_UART_READY_TO_READ_IND
Trace	299	1636	15:33:15.360 ...	MOD_EAT			eat_core_uart.c:62:when-1 len=0
Trace	300	1636	15:33:15.360 ...	MOD_EAT			can_write_len:1
Trace	301	1636	15:33:15.360 ...	MOD_EAT			tail-11 head-11
Trace	302	1636	15:33:15.360 ...	MOD_EAT			main.c:292:msg 7
Trace	303	1636	15:33:15.360 ...	MOD_EAT			get_uart0 buf.
Primitive	304	1636	15:33:15.360 ...	MOD_EAT	MOD_ATCI	INVALID_SAP	MSG_ID_UART_READY_TO_READ_IND
Primitive	305	1637	15:33:15.360 ...	MOD_ATCI	MOD_EAT	INVALID_SAP	MSG_ID_UART_READY_TO_READ_IND
Primitive	306	1637	15:33:15.360 ...	MOD_ATCI	MOD_DS	INVALID_SAP	MSG_ID_READY_UART_WRITE_IND
Trace	307	1637	15:33:15.360 ...	MOD_EAT			main.c:292:msg 4