Modern Odometer Manipulation

DEFCON 33 CHV / collin



Agenda

- **About Me**
  Quick intro
- **Odometer Basics**
  How and why?
- **Modern Manipulation**
  Reverse Engineering!
- **Who cares?**
  Besides us used car buyers!

Disclaimer

Odometer Fraud is a Federal Crime under Title 49 U.S. Code 32703

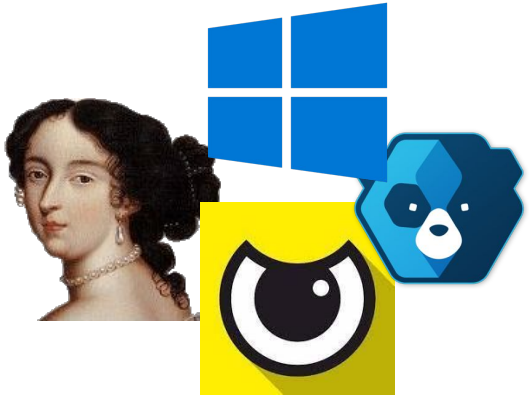
Odometer manipulation devices are illegal to use or advertise or sell

Do not manipulate any odometers

This presentation is for educational purposes only

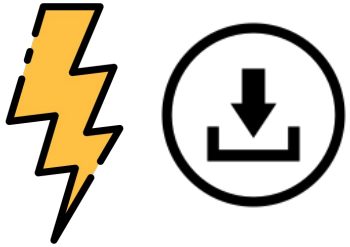
About Me

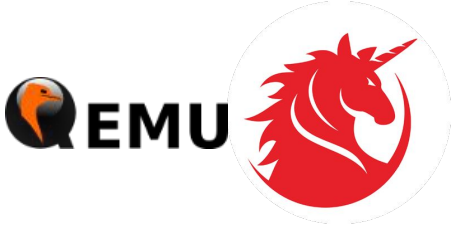




Always liked to hack things!

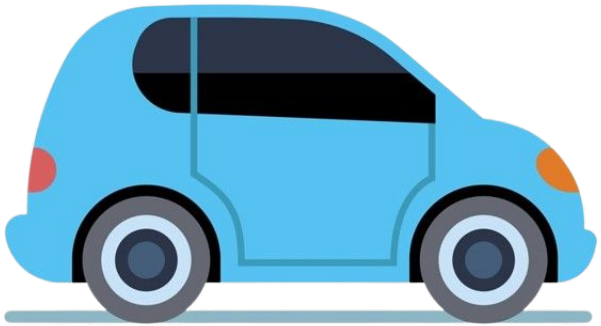
- ➔ iOS Jailbreaking
- ➔ Video Game Hacking
- ➔ Voltage Fault Injection
- ➔ MCU Emulation

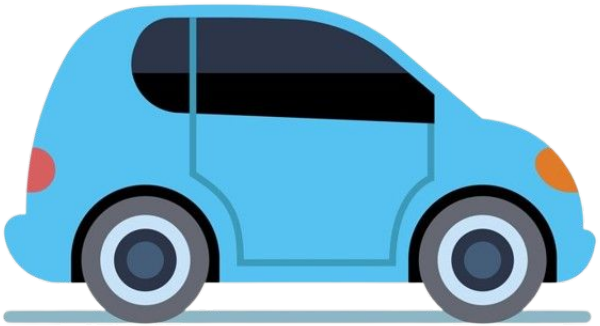




Imagine Two Identical Cars

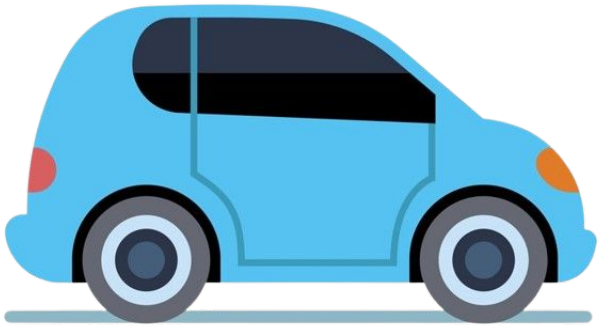
Same year, make and model





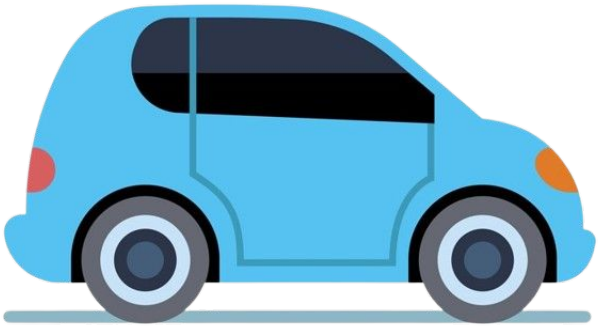
Which is worth more?

The Odometer Drives Value



30,000 miles

$$$



100,000 miles

$

Odometer Represents Wear & Tear



Odometer should only INCREMENT



Modern Manipulation

discussion of a device caught my attention on an automotive forum

You cannot easily roll back the odometer on modern cars

Understanding

Can you prevent an odometer from increasing in the first place?

Idea



Web Store!?

These devices were for sale online:

- ➔ **53+ Makes**
- ➔ **Countless Models**
- ➔ **Configurable Modes**
- ➔ **Bluetooth App Control**
- ➔ **€300 - €600**
- ➔ **USA Shadow Site**

I bought one...

Each device supports one make and model



Man in the Middle

Full control of messages in either direction



Components





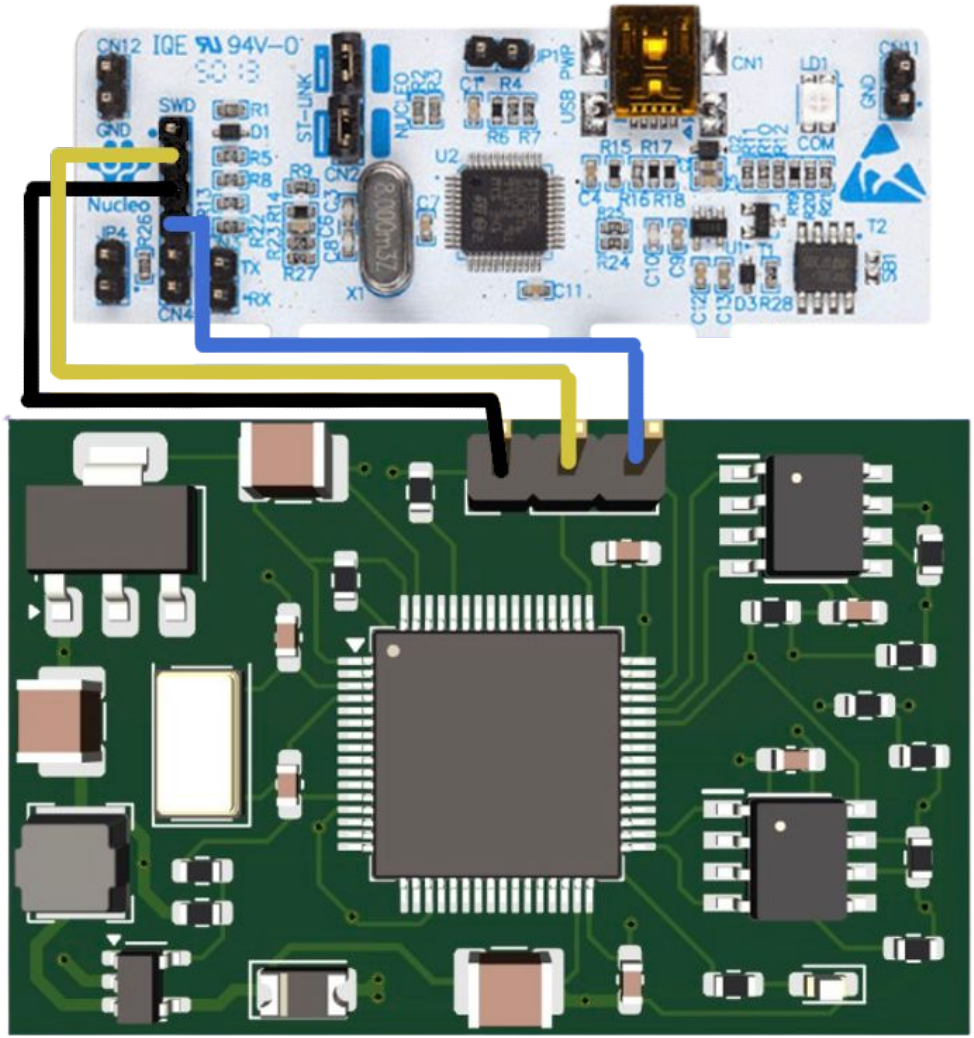
My Goal

Understand how this device works!

- ➔ **Obtain Device's Code**
- ➔ **STM32 only NVM**
- ➔ **SWD?**

SWD

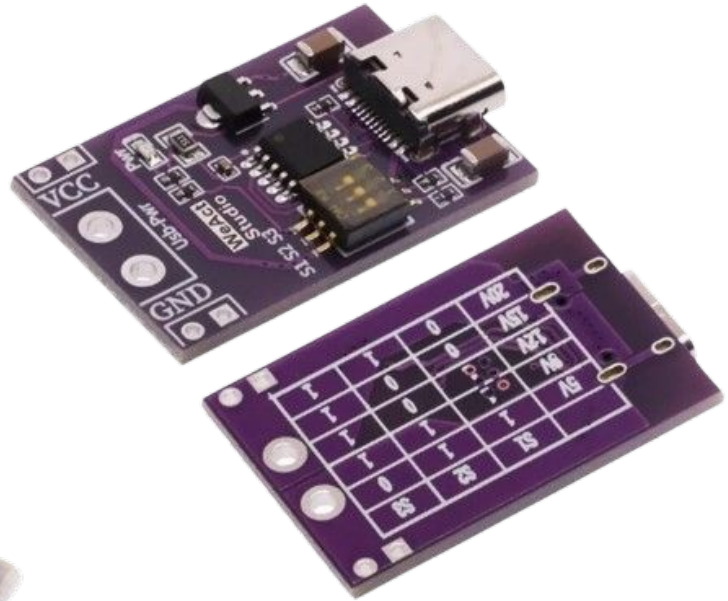
Connecting to SWD



Power

USB-C PD Decoy





OpenOCD

Connection successful!

Can read registers...

```
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
Info : Listening on port 3333 for gdb connections
target halted due to debug-request, current mode:
Handler HardFault
xPSR: 0x01000003 pc: 0x0800016e msp: 0x20000a88
```

```
(gdb) i r
r0             0x0                 0
r1             0x40003000          1073754112
r2             0x20000138          536871224
r3             0x8009800           134256640
r4             0x20000004          536870916
r5             0x1                 1
r6             0x0                 0
r7             0x4725d07d          1193660541
               0x31205808          824203272
               0x2d9eb0c6          765374662
               0x2088ff01          545849089
               0x92054213          -1845149165
               0x44b84444          1152926788
               0x20000a88          0x20000a88
               0xfffffff9          -7
pc             0x800016e           0x800016e
```

OpenOCD

Flash reads all zero,

RAM reads data!

```
(gdb) x/16x 0x8000000
0x8000000:      0x00000000      0x00000000      0x00000000      0x00000000
0x8000010:      0x00000000      0x00000000      0x00000000      0x00000000
0x8000020:      0x00000000      0x00000000      0x00000000      0x00000000
0x8000030:      0x00000000      0x00000000      0x00000000      0x00000000
```

```
(gdb) hexdump 0x20000000 0x100
00000000  02 00 00 00 04 00 00 00  00 00 00 00 01 01 00 00  |................|
00000010  00 00 00 00 05 01 0b 00  1a 00 00 00 00 00 00 00  |................|
00000020  00 00 00 00 00 00 00 00  95 1b 00 00 00 00 00 00  |................|
00000030  c1 1a 00 00 e8 f7 ff 1f  00 80 00 08 00 88 00 08  |................|
00000040  01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000050  ff ff ff ff 00 00 00 00  00 00 00 00 01 00 00 00  |................|
00000060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000070  00 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00  |................|
00000080  00 00 24 30 2a 30 30 30  30 30 30 30 5f 24 31 2a  |..$0*0000000_$1*|
00000090  30 30 25 53 4f 4e 5f 5f  24 32 2a 32 35 25 53 4f  |00%SON__$2*25%SO|
000000a0  4e 5f 5f 24 33 2a 31 30  25 53 4f 4e 5f 5f 24 34  |N__$3*10%SON__$4|
000000b0  2a 39 39 25 46 52 45 45  5f 24 35 2a 52 45 53 45  |*99%FREE_$5*RESE|
000000c0  52 56 45 5f 24 36 2a 52  45 53 45 52 56 45 5f 24  |RVE_$6*RESERVE_$|
000000d0  37 2a 52 45 53 45 52 56  45 5f 24 38 2a 52 45 53  |7*RESERVE_$8*RES|
000000e0  45 52 56 45 5f 24 39 2a  52 45 53 45 52 56 45 5f  |ERVE_$9*RESERVE_|
000000f0  95 1b 00 00 00 a2 4a 04  00 64 00 40 04 00 00 00  |......J..d.@....|
```

OpenOCD

Read protection is on!

```
> stm32f1x options_read 0
option byte register = 0x3ffffe
write protection register = 0xffffffff
read protection: on
watchdog: software
```

Time to do some research...



Query

STM32F1 flash protection bypass vulnerability read out

Promising result!

Shellcode Exec. via Glitch and FPB is explained in the paper One Exploit to Rule them All? by Johannes Obermaier, Marc Schink and Kosma Moczek

Flash Protection

ON in three conditions:

- **Debug probe connected**
- **Boot to bootloader**
- **Boot to SRAM**

The latter two conditions are cleared at reset, the first only on power cycle!

Shellcode Exec. via Glitch and FPB

1 **Prepare SRAM**

With SWD load two stage payload into SRAM, then disconnect debugger

2 **Power Cycle**

Clear debugger attached Flash Protection condition, abuse SRAM retention

3 **Boot to SRAM**

Assert BOOT0 pin to boot to stage 1 in SRAM

4 **Setup FPB**

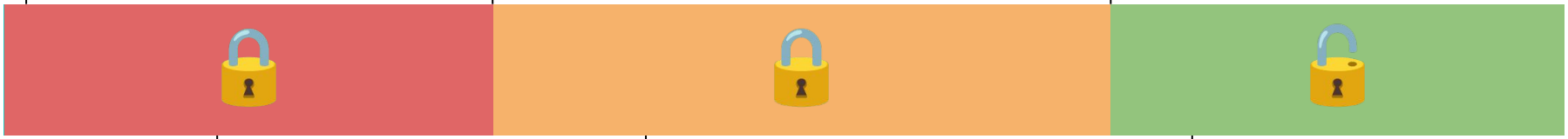
Setup Flash Patch Block on Flash entry point address, point to stage 2

5 **Boot to Flash**

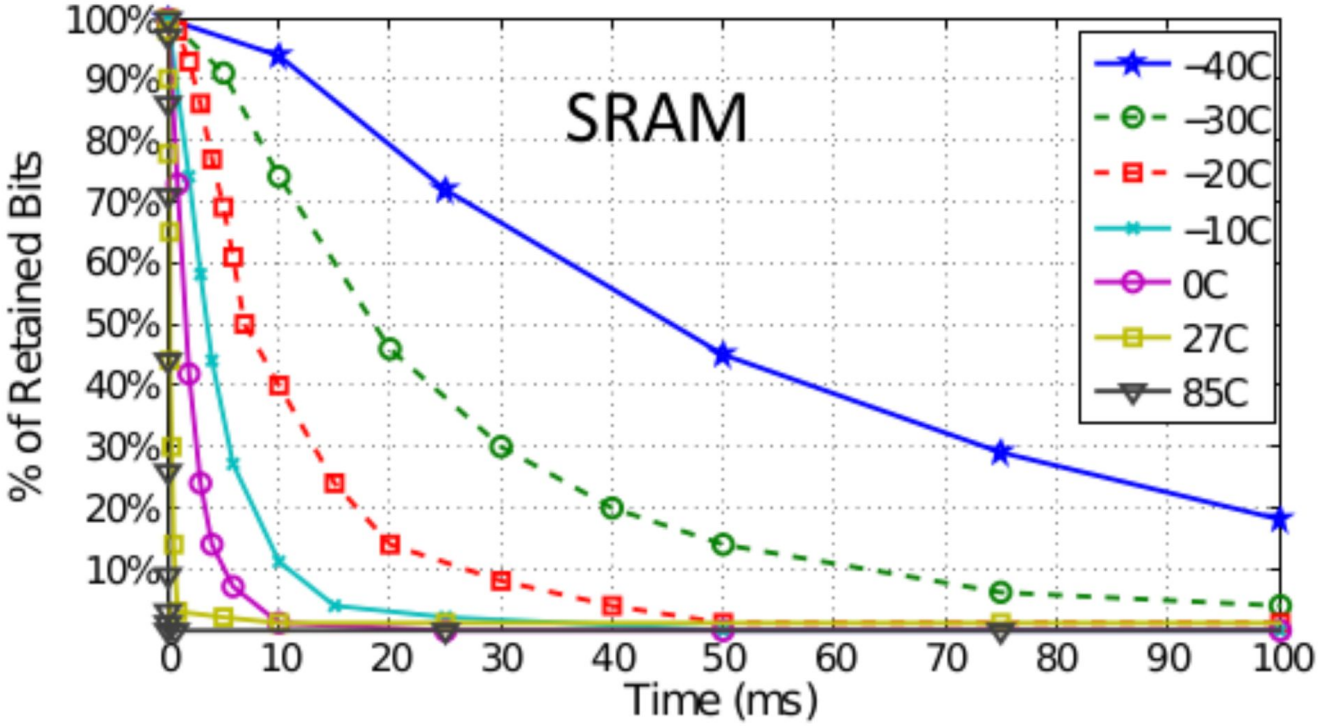
De-assert BOOT0 pin to boot to Flash, FPB will redirect to stage 2

6 **Flash Dump**

Stage 2 transmits Flash contents over UART



SRAM Data Retention



Implementation?

CTXz made it easy!



`CTXz/stm32f1-picopwner`

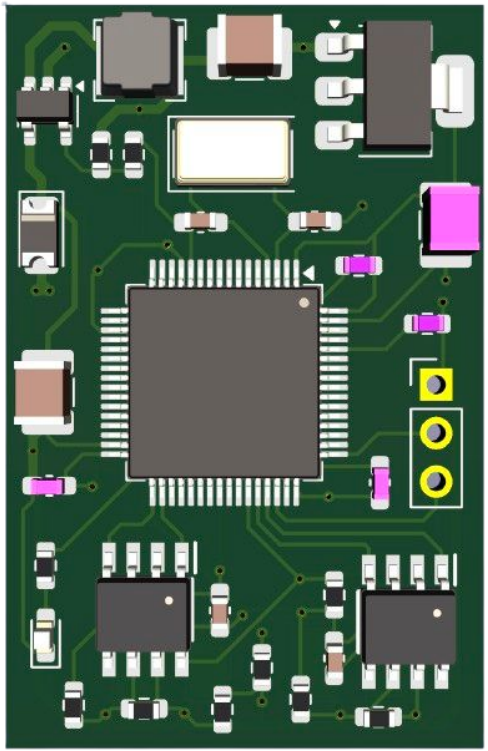
- Usage
  - What you'll need to get started
  - Pre-requisites
  - Cloning the repository
  - Building and flashing the attack firmware onto the Pi Pico
  - Building the target board exploit firmware
  - Hardware Setup
  - Executing the attack
  - Troubleshooting
- How does the attack work?
  - Relevant STM32F1 properties
    - Read-Out Protection (RDP)
    - The Flash Patch and Breakpoint Unit (FPB)
    - SRAM
  - 2-Stage Exploit Firmware
    - Stage 1
    - Stage 2
  - The Attack
    - Step 1: Preparation
    - Step 2: Power Glitching
    - Step 3: Exploit Firmware - Stage 1
    - Step 4: Exploit Firmware - Stage 2

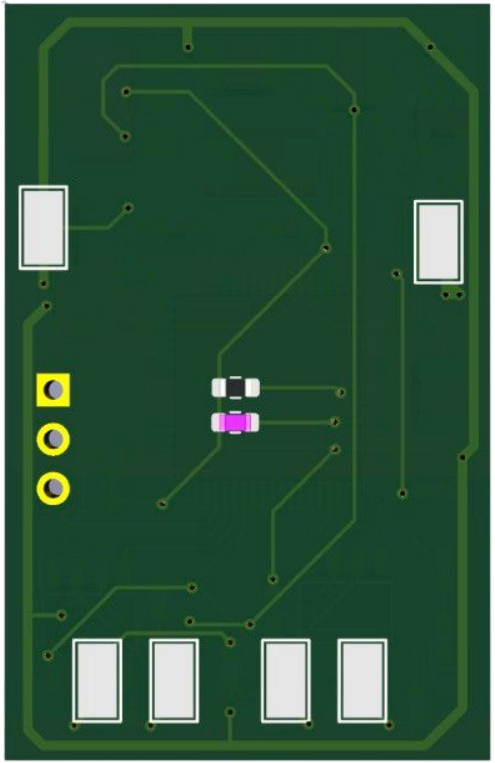
Modifying the Hardware

Control VCC

Observe nRST

Control BOOTCFG





Instrumentation

VCC

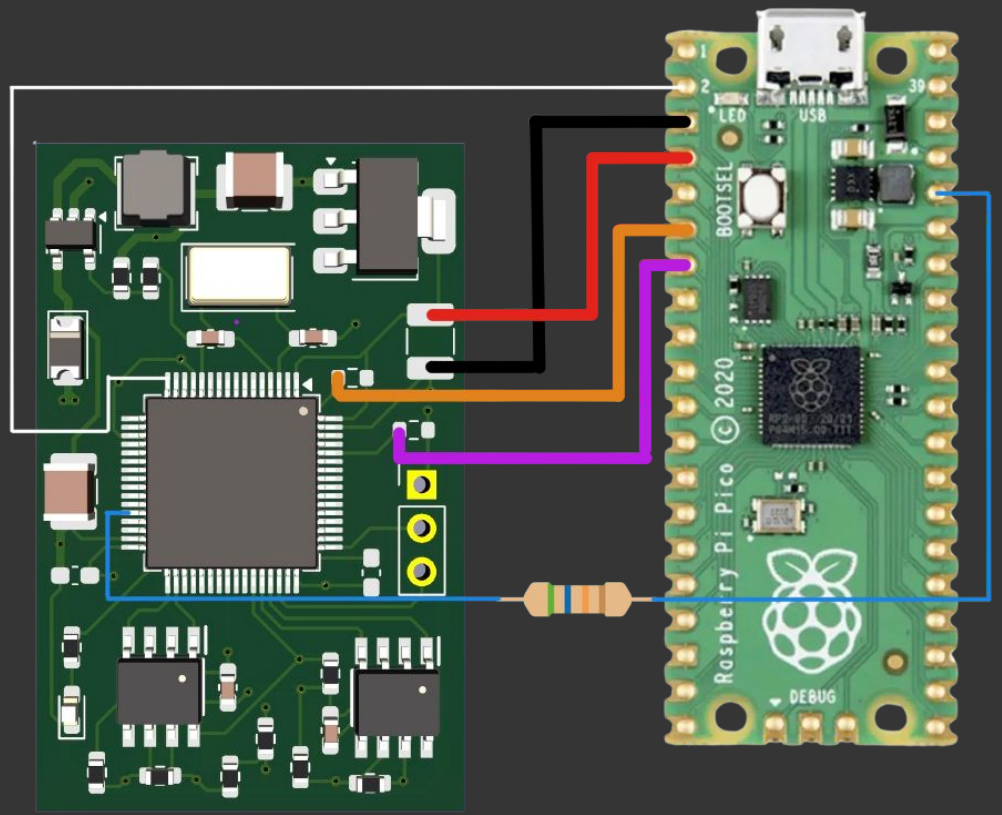
GND

nRST

UART TX

BOOT0

BOOT1

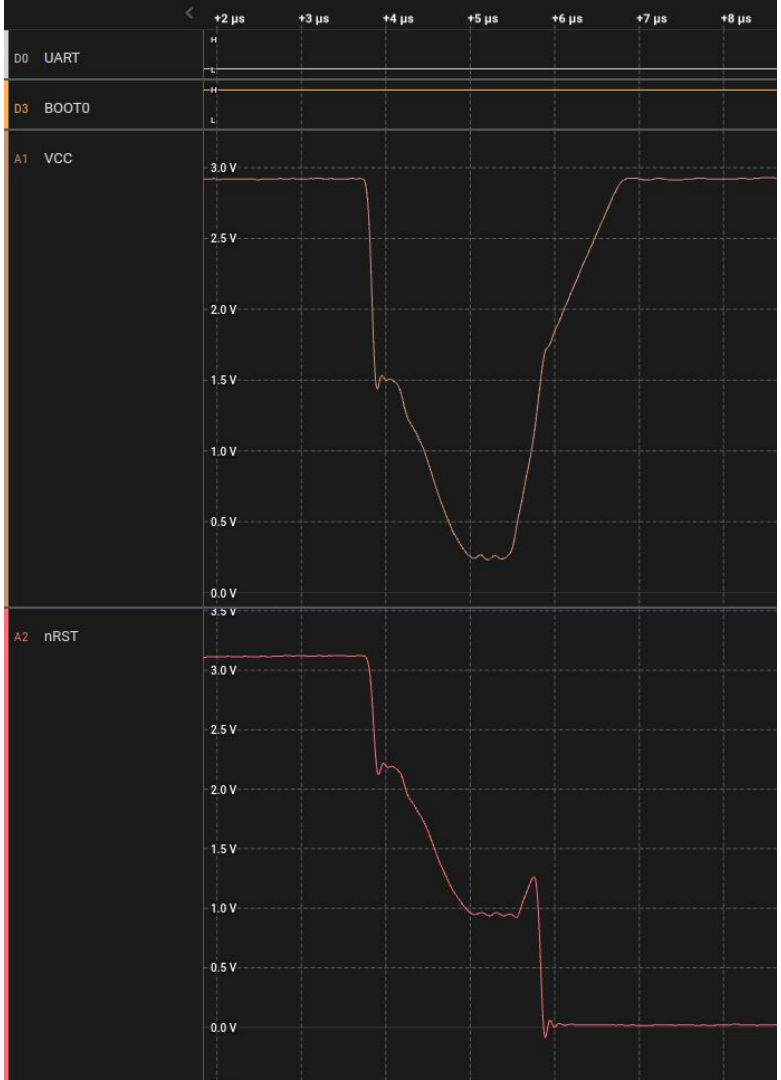


Running the Attack

VCC = 0

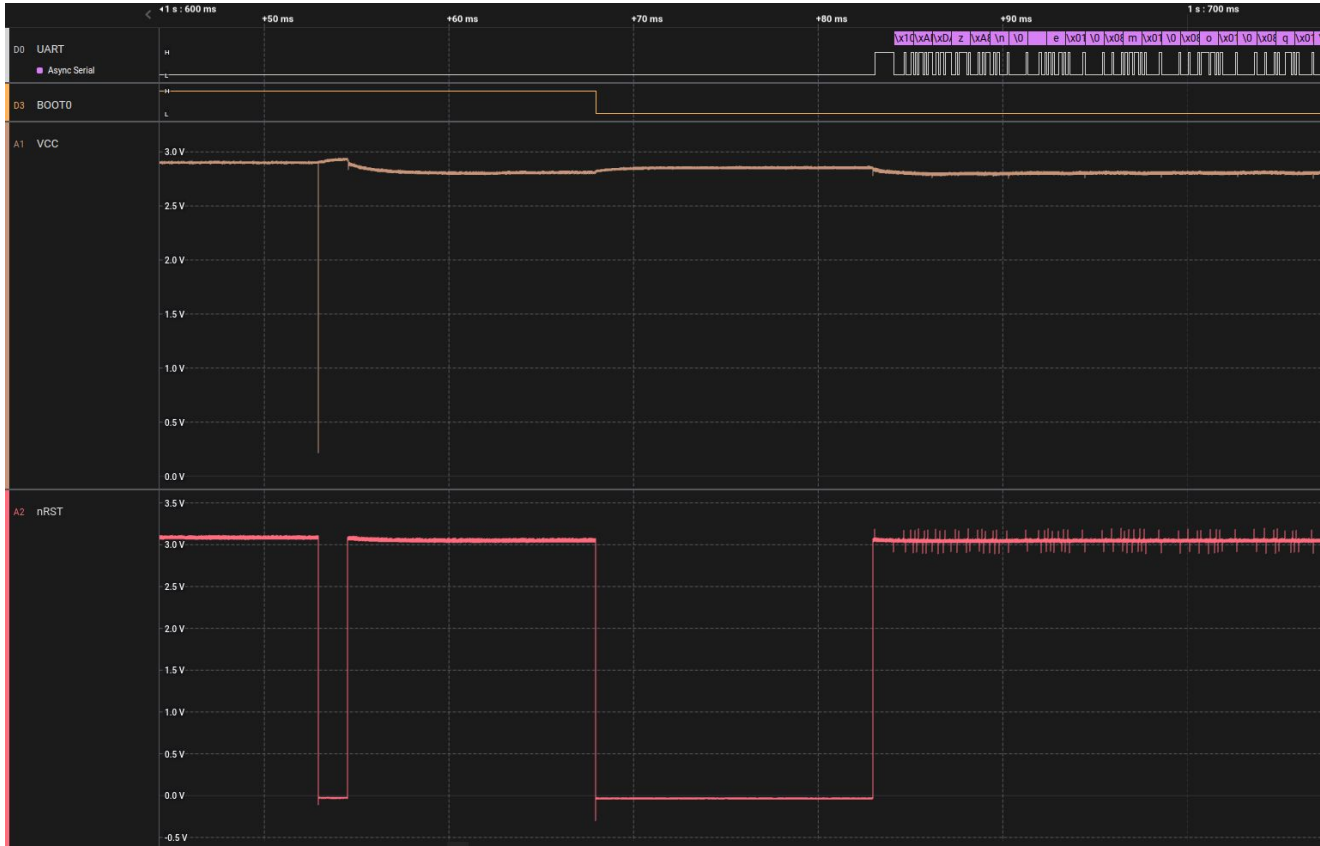
while nRST {}

VCC = 1



Success

Flash obtained!



Finally! Reverse Engineering



Next

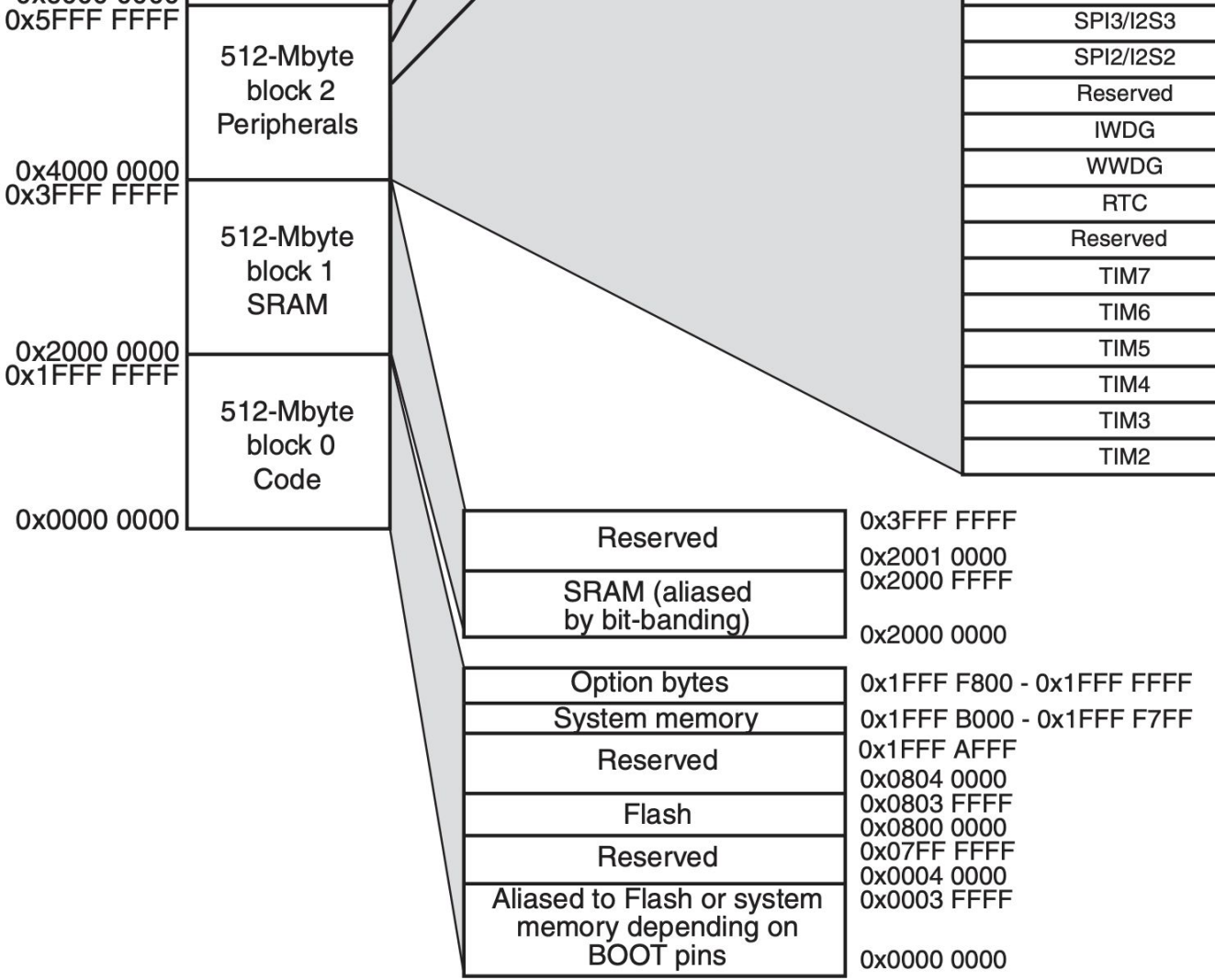
Let's get into Ghidra!

Memory

0x20000000 SRAM

0x08000000 Flash

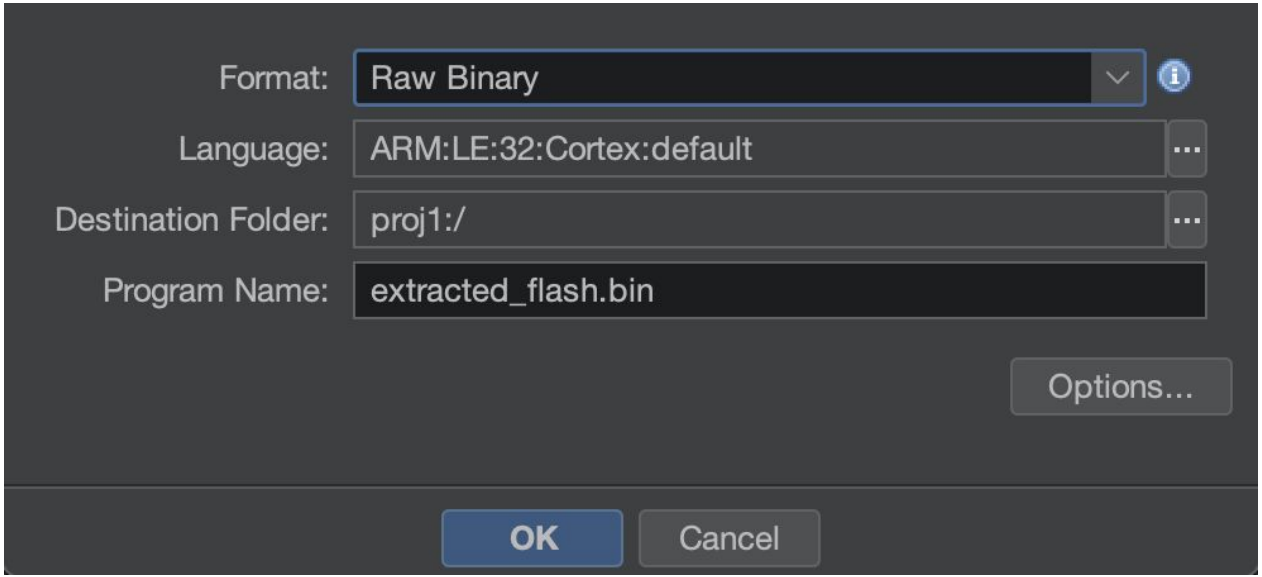
0x0 Flash Mirror



Ghidra Load

ARM Cortex Little Endian





SVD Loader

Automatically sets up:

Memory Map
Peripheral Register Types



leveldown-security/SVD-Loader-Ghidra

```
Generating memory blocks...
          TIM2_TIM3_TIM4_TIM5_TIM6_TIM7_TIM12_TIM13_TIM14(0x40
          RTC_WWDG_IWDG(0x40002800:0x40003400)
          SPI2_SPI3(0x40003800:0x40004000)
          USART2_USART3_UART4_UART5_I2C1_I2C2_USB(0x40004400:0
          CAN1_CAN2(0x40006400:0x40006c00)
          BKP_PWR_DAC(0x40006c04:0x40007800)
          AFIO_EXTI_GPIOA_GPIOB_GPIOC_GPIOD_GPIOE_GPIOF_GPIOG_
          TIM9_TIM10_TIM11(0x40014c00:0x40015800)
          SDIO(0x40018000:0x40018400)
          DMA1_DMA2(0x40020000:0x40020800)
          RCC(0x40021000:0x40021400)
          FLASH(0x40022000:0x40022400)
          CRC(0x40023000:0x40023400)
          Ethernet_MAC(0x40028000:0x40028061)
          Ethernet_MMC(0x40028100:0x40028500)
          Ethernet_PTP(0x40028700:0x40028b00)
          Ethernet_DMA(0x40029000:0x40029400)
          OTG_FS_GLOBAL_USB_OTG_HOST_OTG_FS_DEVICE(0x50000000:
          OTG_FS_PWRCLK(0x50000e00:0x50001200)
          FSMC(0xa0000000L:0xa0001000L)
          SCB_ACTRL(0xe000e008L:0xe000e00dL)
          STK(0xe000e010L:0xe000e021L)
          NVIC(0xe000e100L:0xe000e43dL)
          SCB(0xe000ed00L:0xe000ed41L)
          MPU(0xe000ed90L:0xe000eda5L)
          NVIC_STIR(0xe000ef00L:0xe000ef05L)
          DBGMCU(0xe0042000L:0xe0042400L)
          Done!
Generating peripherals...
          PWR
                    0x40007000:0x40007008
                              CR(0x0:0x4)
                              CSR(0x4:0x8)
```

SVD Loader

Decompile: FUN_08000cd0 - (extracted_flash.bin)

```
undefined4 FUN_08000cd0(int *param_1,uint *param_2)

{
  uint uVar1;
  uint uVar2;

  uVar1 = 1 << *(sbyte *)(param_2 + 5);
  *(uint *)(*param_1 + 0x200) = *(uint *)(*param_1 + 0x200) & 0xffffc0ff | p
  *(uint *)(*param_1 + 0x21c) = *(uint *)(*param_1 + 0x21c) & ~uVar1;
  if (param_2[7] == 0) {
    *(uint *)(*param_1 + 0x20c) = *(uint *)(*param_1 + 0x20c) & ~uVar1;
    *(uint *)(*param_1 + param_2[5] * 8 + 0x240) =
         param_2[1] & 0xffff | (uint)*(ushort *)(param_2 + 3) << 0x10;
    *(uint *)(*param_1 + param_2[5] * 8 + 0x244) =
         *param_2 & 0xffff | (uint)*(ushort *)(param_2 + 2) << 0x10;
  }
  if (param_2[7] == 1) {
    *(uint *)(*param_1 + 0x20c) = *(uint *)(*param_1 + 0x20c) | uVar1;
    *(uint *)(*param_1 + param_2[5] * 8 + 0x240) =
         param_2[1] & 0xffff | (uint)*(ushort *)param_2 << 0x10;
    *(uint *)(*param_1 + param_2[5] * 8 + 0x244) =
         param_2[3] & 0xffff | (uint)*(ushort *)(param_2 + 2) << 0x10;
  }
  uVar2 = *(uint *)(*param_1 + 0x204);
  if (param_2[6] == 0) {
```

Decompile: can_configure_filter - (od.bin)

```
undefined4 can_configure_filter(comms_CAN *comms,can_filter_con

{
  uint mailbox_bit;
  uint uVar1;

  mailbox_bit = 1 << *(sbyte *)&config->mailbox_index;
  comms->CAN->FMR = comms->CAN->FMR & 0xffffc0ff | config->filt
  comms->CAN->FA1R = comms->CAN->FA1R & ~mailbox_bit;
  if (config->filter_bit_mode == 0) {
    comms->CAN->FS1R = comms->CAN->FS1R & ~mailbox_bit;
    (&comms->CAN->FB0_FR1)[config->mailbox_index * 2] =
         config->filters1_lower & 0xffff | (uint)*(ushort *)&co
    (&comms->CAN->FB0_FR2)[config->mailbox_index * 2] =
         config->filters2_lower & 0xffff | (uint)config->filter
  }
  if (config->filter_bit_mode == 1) {
    comms->CAN->FS1R = comms->CAN->FS1R | mailbox_bit;
    (&comms->CAN->FB0_FR1)[config->mailbox_index * 2] =
         config->filters1_lower & 0xffff | (uint)*(ushort *)&co
    (&comms->CAN->FB0_FR2)[config->mailbox_index * 2] =
         config->filters1_upper & 0xffff | (uint)config->filter
  }
  uVar1 = comms->CAN->FM1R;
```

Entry Point

Well Defined in ARM

```
                    //
                    // flash_mirror
                    // ram:00000000-ram:0003ffff
                    //

     assume spsr = 0x0  (Default)
                    MasterStackPointer
00000000 a8 0a 00 20     addr        DAT_20000aa8

                    Reset
00000004 65 01 00 08     addr        FUN_08000164+1

                    NMI
00000008 6d 01 00 08     addr        DAT_0800016d

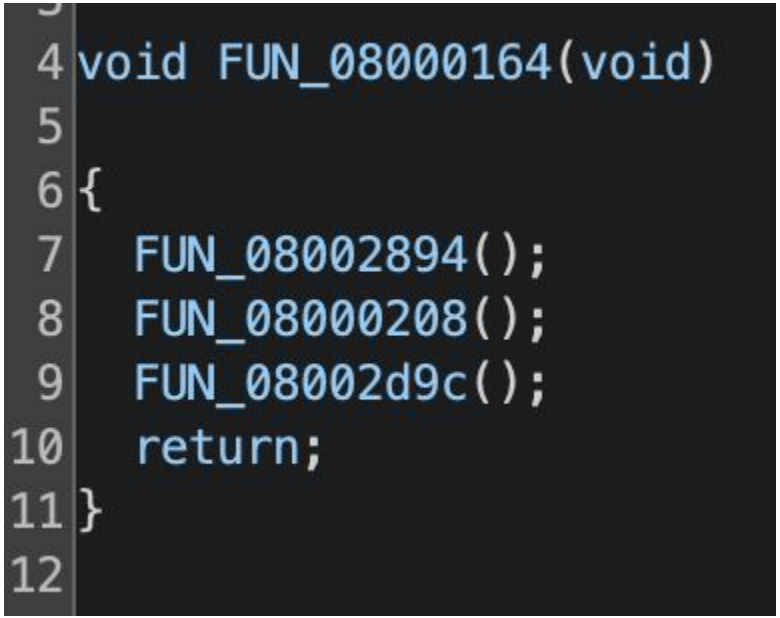
                    HardFault
0000000c 6f 01 00 08     addr        DAT_0800016f

                    MemManage
00000010 71 01 00 08     addr        DAT_08000171

                    BusFault
00000014 73 01 00 08     addr        DAT_08000173
```

Finding Functionality

Follow last call



```
4  void FUN_08000164(void)
5
6  {
7    FUN_08002894();
8    FUN_08000208();
9    FUN_08002d9c();
10   return;
11 }
12
```

Following

Follow last call

```
1
2  void FUN_08002d9c(void)
3
4  {
5    FUN_080016bc();
6    FUN_0800281c();
7    FUN_08002680();
8    FUN_08002730();
9    DAT_20000178 = &Peripherals::IWDG;
10   DAT_2000017c = 2;
11   DAT_20000180 = 1000;
12   FUN_080015d0();
13   FUN_08001680(&DAT_20000178);
14   FUN_080025e0();
15   FUN_08002630();
16   FUN_08000318();
17   do {
18                     /* WARNING: Do nothin
19   } while( true );
20 }
21
```

Following

Last call is in a loop!

```
1
2  void FUN_08000318(void)
3
4  {
5    DAT_20000128 = &DAT_200001c8;
6    DAT_2000012c = &DAT_20000240;
7    DAT_20000168 = &DAT_200001e4;
8    DAT_2000016c = &DAT_2000021c;
9    FUN_080015b8(&Peripherals::GPIOB,0x80,0);
10   FUN_08000c40();
11   FUN_080013e6(5);
12   DAT_20000050 = FUN_08002d30(DAT_2000003c);
13   DAT_20000040 = FUN_08002d30(DAT_20000038);
14   if (DAT_20000040 == -1) {
15     DAT_20000040 = 1;
16     FUN_08002cb0(DAT_20000038);
17   }
18   DAT_2000000d = (byte)DAT_20000040 & 0xf;
19   FUN_08000928();
20   FUN_080027e4();
21   FUN_080015b8(&Peripherals::GPIOA,0x400,0);
22   FUN_080011b4(&DAT_200000f8,0);
23   FUN_080011b4(&DAT_20000138,1);
24   DAT_2000000c = 5;
25   do {
26     FUN_0800022c();
27   } while( true );
28 }
```

Loop

Main Functionality

```
void app_loop(void)

{
  uint now;
  int now2;

  app_loop_start_ticks = get_ticks();
  now = get_ticks();
  if (last_fwd_msg_ticks < now) {
    now2 = get_ticks();
    if ((10000 < now2 - last_fwd_msg_ticks) && (last_fwd_msg_ticks != 0)) {
      last_fwd_msg_ticks = 0;
      DAT_20000009 = 1;
      bluetooth_enabled = 0;
                    /* if its been a while since we forwarded a msg, turn bluetooth off */
      disable_bluetooth();
    }
  }
                    /* hook messages, perform MITM */
  can_hook_car_to_ipc();
  can_hook_ipc_to_car();
  refresh_watchdog((IWDG *)&pIWDG);
  if (DAT_20000007 == '\x01') {
    DAT_20000007 = '\0';
    uart2_send();
  }
  if (pending_config_changed_notif == '\x01') {
    pending_config_changed_notif = '\0';
    send_config_changed_notif();
  }
  uart2_command_handler();
```

Manipulation

Odometer hooks

```
}
                /* BO_ 377 EngineData_7 (from commai opendbc)
                     has property: OdoCount */
if ((saved_frame0.standard_id == 0x179) && (saved_frame0.dlc == 8)) {
  if (DAT_2000000d == 1) {
                /* in mode 1, zero out odometer increment value */
    saved_frame0.data[5] = 0;
  }
  else {
                /* modify forwarded odometer amount based on selected config mode */
    if (DAT_2000000d == 2) {
                /* in mode 2, divide odometer value by 4 */
      modify_odometer_value
                (auStack_34,0x179,saved_frame0.extended_id,saved_frame0.is_extended,
                 saved_frame0.is_remote,8,saved_frame0.data._0_4_,saved_frame0.data._4_4_,4);
      memcpy(&saved_frame0,auStack_34,0x1c);
    }
    if (DAT_2000000d == 3) {
                /* in mode 3, divide odometer value by 6 */
      modify_odometer_value
                (auStack_34,saved_frame0.standard_id,saved_frame0.extended_id,
                 saved_frame0.is_extended,saved_frame0.is_remote,saved_frame0.dlc,
                 saved_frame0.data._0_4_,saved_frame0.data._4_4_,6);
      memcpy(&saved_frame0,auStack_34,0x1c);
    }
  }
}
```

Controls

Change mode via steering wheel buttons!

```
iVar1 = get_ticks();
if ((6000 < (uint)(iVar1 - btn_last_pressed_time)) && (btn_last_pressed_time != 0)) {
  btn_last_pressed_time = 0;
  steering_btn_held_time = 0;
}
if (saved_frame0.is_extended == 0) {
                    /* BO_ 129 Steering_Wheel_Data (from commai opendbc)
                         has properties for each button on the wheel */
  if ((saved_frame0.dlc == 8) &&
     ((saved_frame0.standard_id == 0x81 || (saved_frame0.standard_id == 0x82)))) {
    if ((saved_frame0.data[0] >> 4 == 1) || ((saved_frame0.data._0_4_ & 0xf) == 4)) {
      btn_last_pressed_time = get_ticks();
      steering_btn_held_time += 1;
      if (steering_btn_held_time == 9) {
                    /* update odometer bypass config mode via steering wheel buttons */
        DAT_2000000d += 1;
        if (4 < DAT_2000000d) {
          DAT_2000000d = 1;
        }
        DAT_2000000f = DAT_2000000d;
      }
      if (saved_frame0.is_extended != 0) goto LAB_08000ba4;
    }
    else {
      steering_btn_held_time = 0;
      btn_last_pressed_time = 0;
    }
  }
```

Config Changed

Notification when config mode updated

```
2  void send_config_changed_notif(void)
3
4  {
5    if (DAT_2000000f != '\0') {
6      notif_delay += 1;
7      if (notif_delay == 1) {
8        out_frame.is_extended = 0;
9        out_frame.standard_id = 0x3b2;
10       out_frame.dlc = 8;
11       out_frame.data[0] = 0x44;
12       out_frame.data[1] = 0x88;
13       out_frame.data[2] = 0x46;
14       out_frame.data[3] = 9;
15       out_frame.data[4] = 0x18;
16       out_frame.data[5] = 3;
17       out_frame.data[6] = 0x40;
18       out_frame.data[7] = 2;
19       pCAN1.frame_out = (candata_ext *)&out_frame;
20       can_send_frame_out(&pCAN1);
21     }
22     if (3 < notif_delay) {
23       notif_delay = 0;
24       DAT_2000000f += -1;
25     }
26   }
27   return;
28 }
29
```

Analysis

Device simply modifies a specific message before it reaches Instrument Panel Cluster!

Who cares if odometers are manipulated?



Meaning

manipulated to display a value lower than it otherwise would.

Car buyers!

But also warranty providers when proving maintenance and lease providers when calculating a residual value!

This includes OEMs!

Threat Model

They thought of rolling back,

did they think of never increasing?

Advice for Buyers

- Pre-purchase inspection
- Consumables wear
- Diagnostic report



Questions?

contact me:
collinrevolt@gmail.com

Bluetooth

Same code without paying for bluetooth!

```
106       else {
107         iVar2 = memcmp(usart2_in_buffer,"OK+NAME:",8);
108         if (iVar2 == 0) {
109           DAT_2000005d = 0;
110           do {
111             uVar1 = (uint)DAT_2000005d;
112             uart_recv_OK_NAME_buf[uVar1] = usart2_in_buffer[uVar1 + 8];
113             DAT_2000005d = (byte)(uVar1 + 1);
114           } while ((uVar1 + 1 & 0xff) != 10);
115           iVar2 = memcmp(uart_recv_OK_NAME_buf,"HMSoft",6);
116           if (iVar2 == 0) {
117             DAT_20000059 = 1;
118           }
119         }
120         else {
121           iVar2 = memcmp(usart2_in_buffer,"OK+CONN",7);
122           if (iVar2 == 0) {
123             bluetooth_device_connected = 1;
124           }
125           else {
126             iVar2 = memcmp(usart2_in_buffer,"OK+LOST",7);
127             if (iVar2 == 0) {
128               bluetooth_device_connected = 0;
129               DAT_2000005e = 0;
130               bluetooth_tx_mode = FIRST_CONFIG;
131             }
```



Questions?

contact me:
collinrevolt@gmail.com