# My First contribution in Linux kernel

and it’s not that hard!

#WMI

#ACPI

#RGB_KEYBOARD

#Reverse_engineering

Presented by: Jafar Akhondali

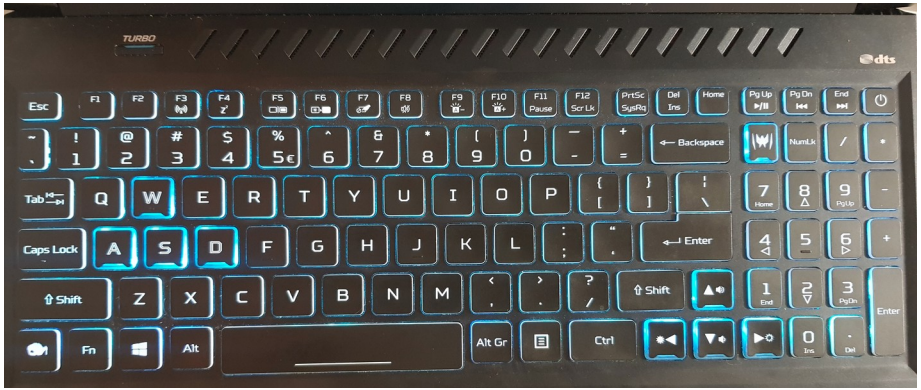
Oct, 2021

# Overview

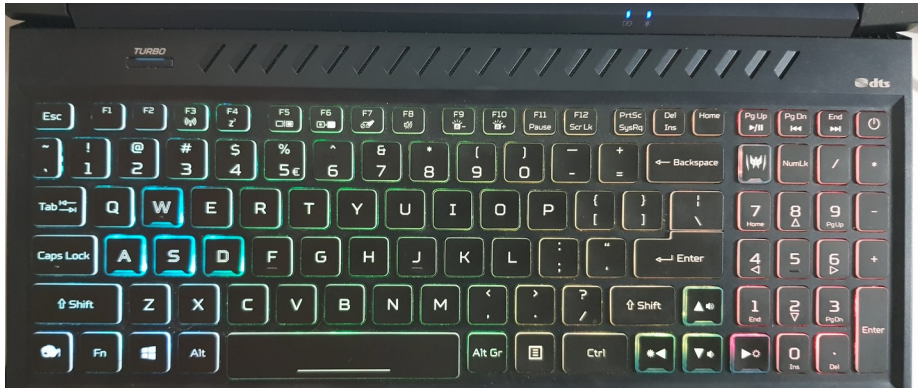
- Problem statement
- Where should I even start?
- How does the low-level operations work
- Implementing the solution in a kernel module
- Submitting the patch (not pull request!)

# What were the problems?

Gnu\Linux



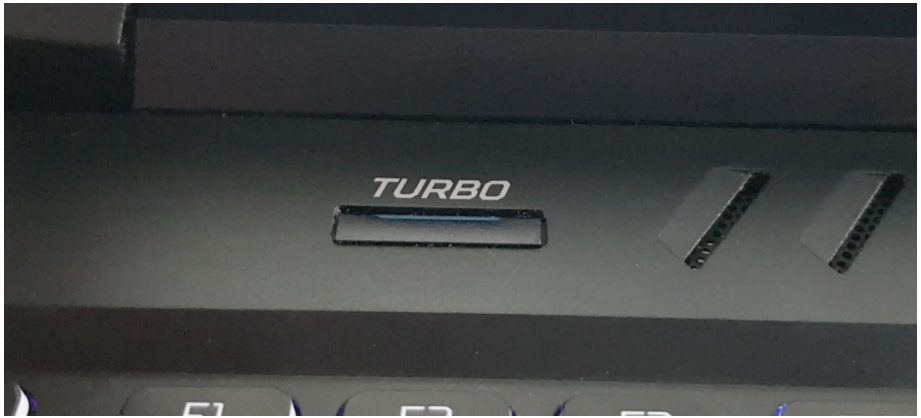

Windows




***Can’t change keyboard color and effects in Linux***

# What were the problems?

Gnu\Linux

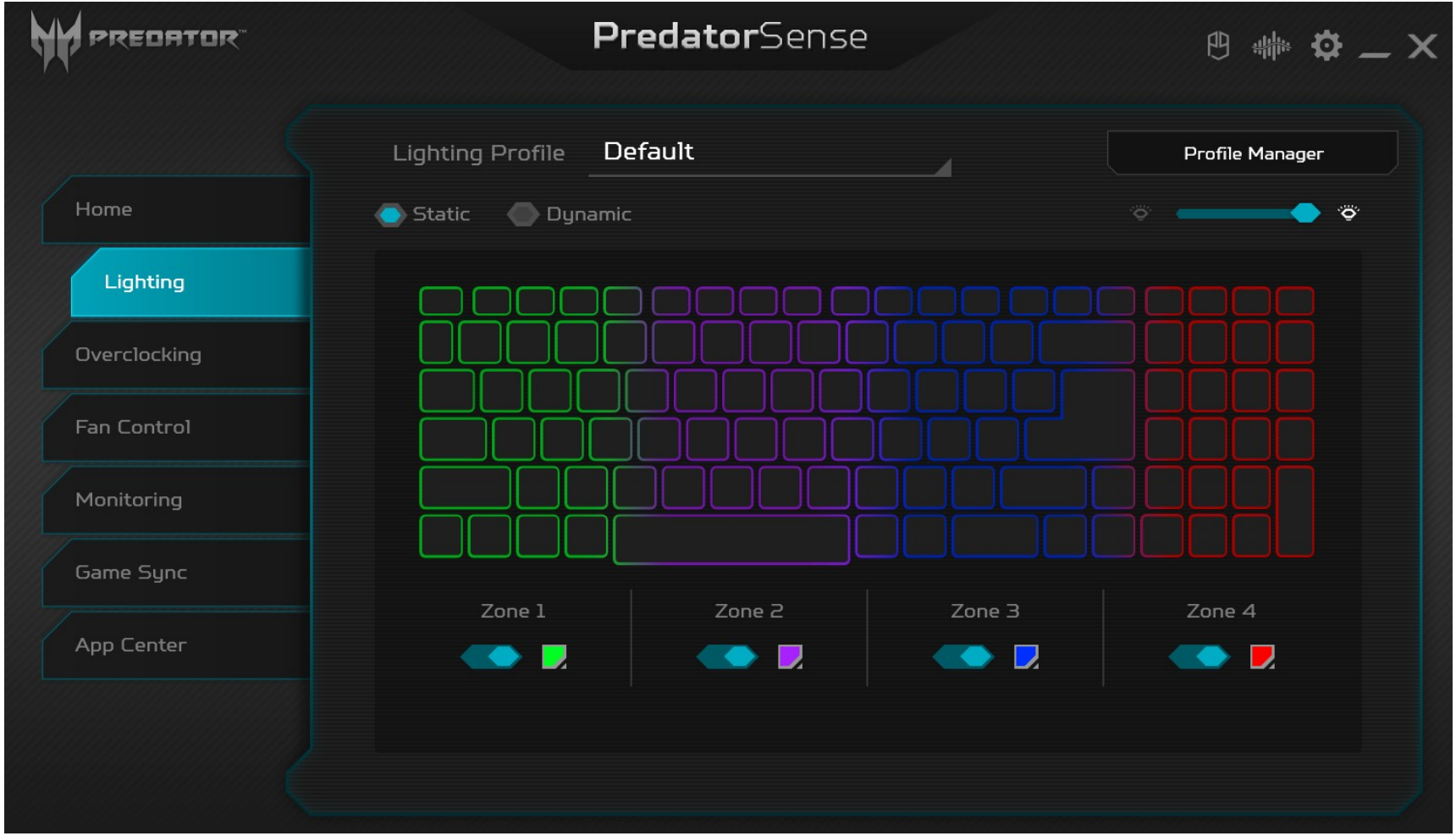



Windows


***Turbo mode only works in Windows***

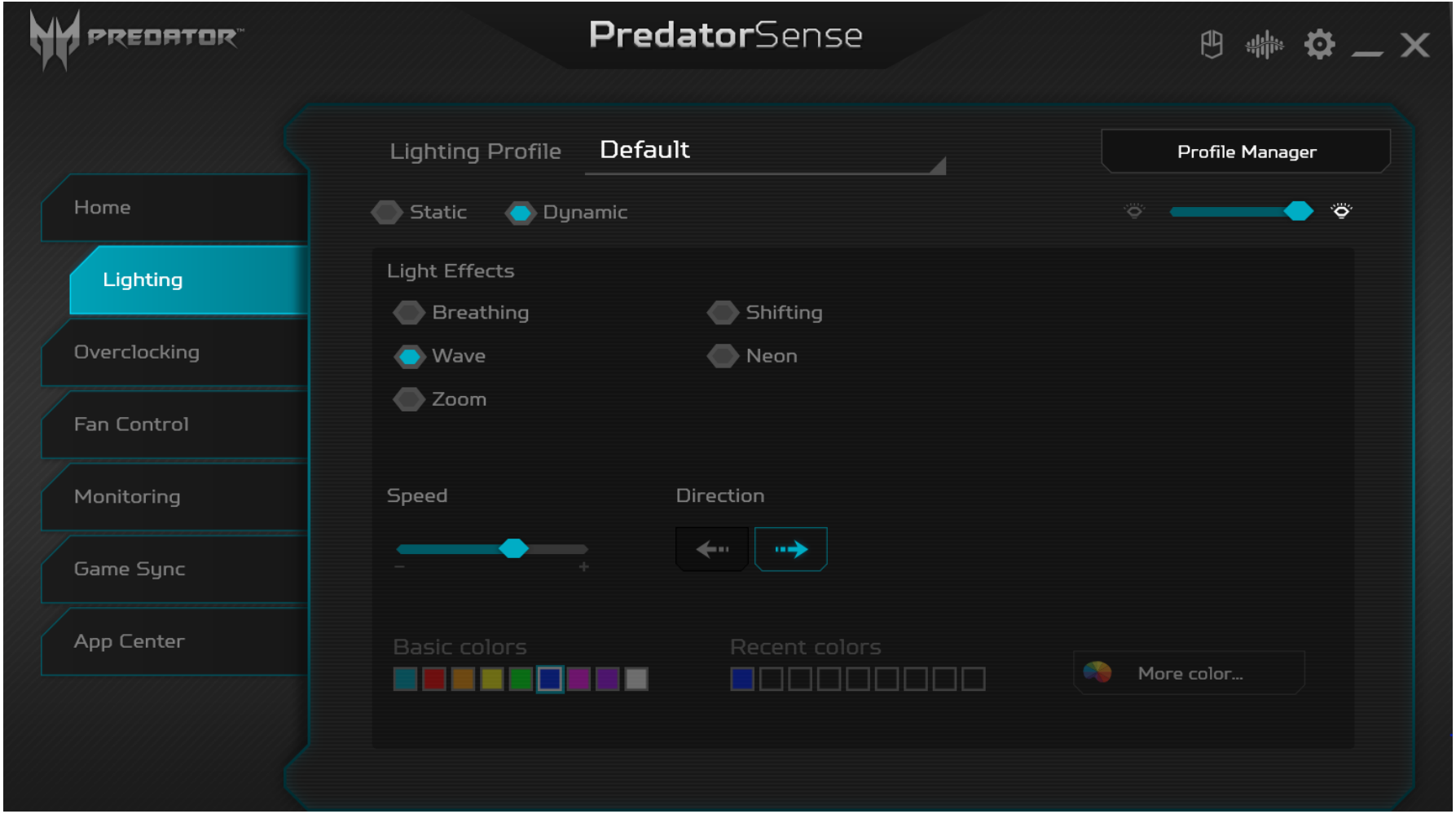
# Predator Sense:

Official Acer software to control software in windows



# Predator Sense:

Official Acer software to control software in windows



# Let's Google it

Reddit:
https://www.reddit.com/r/linuxhardware/comments/jemxq2/linux_support_on_acer_predator_helios_300_2020/

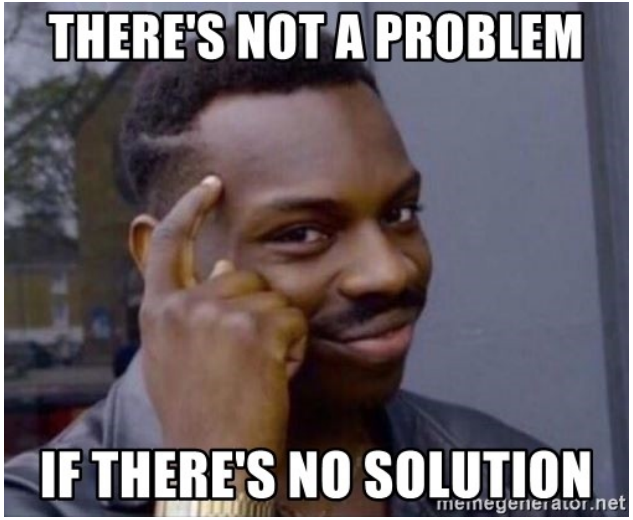
Acer Community:
https://community.acer.com/en/discussion/610885/acer-predator-helios-300-ubuntu-20-04-rgb-keyboard-backlight

AskUbuntu:
https://askubuntu.com/questions/1227139/how-to-enable-the-turbo-button-on-acer-predator-helios-300-in-ubuntu/1357954#1357954

1~3 year old threads

https://community.acer.com/en/discussion/632698/ph315-52-how-to-control-keyboard-backlight-on-linux

Github: No

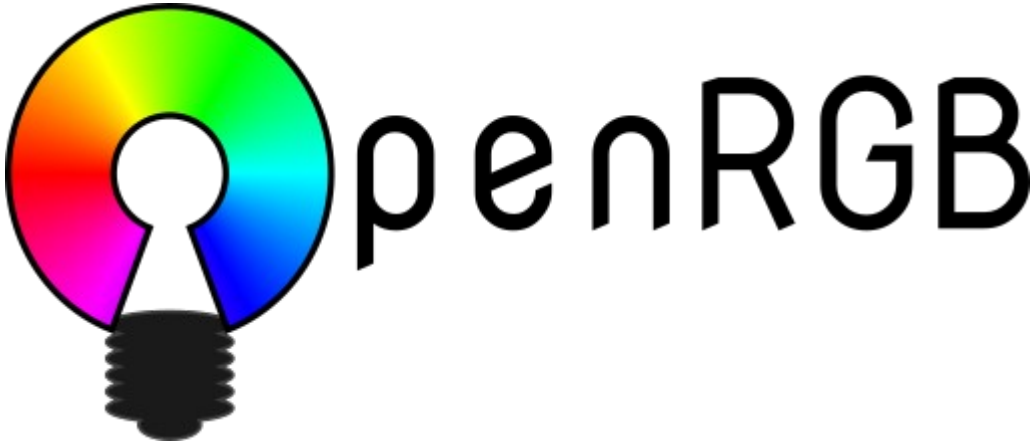



I just ignored keyboard colors for ~2 months

but ...

I do really LOVE RGB

# OpenRGB




Open source RGB lighting control that doesn't depend on manufacturer software.

Opened issue on their Gitlab repository

No answers :-(

# Other solutions?

- Install Predator Sense on a virtual machine Windows image? Didn’t work.
- Install Power shell on Windows? (we’ll cover it later), but didn’t help
- ?

## Let’s begin

At this state, I was sure no one will implement this soon in Gnu\Linux.

But can I do this?

**Problems:**

1) I’m not a C expert
2) I don’t know how these low-level operations work
3) I don’t know how to write a driver
4) I don’t even know the list of things I need to know

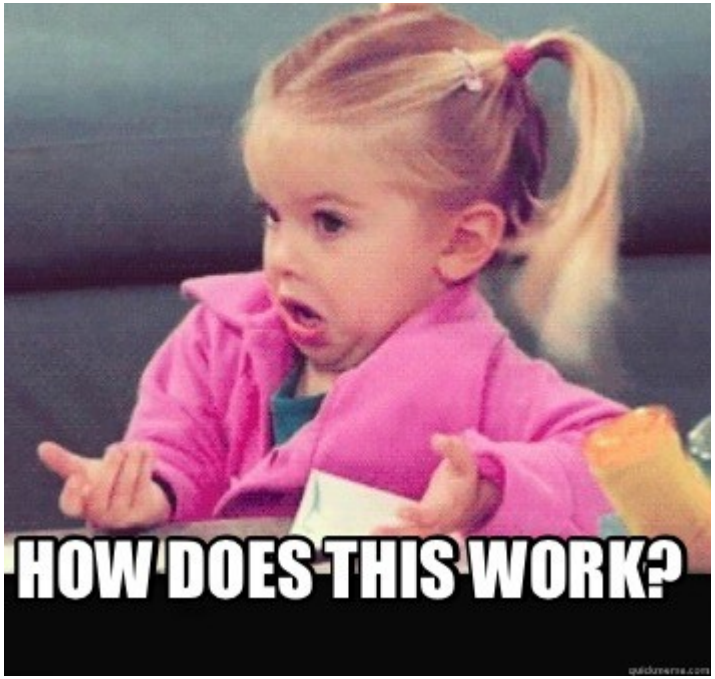
# How does Predator Sense controls the hardware?!

**First idea:**

OpenRGB mostly tries to implement data transferred by USB.

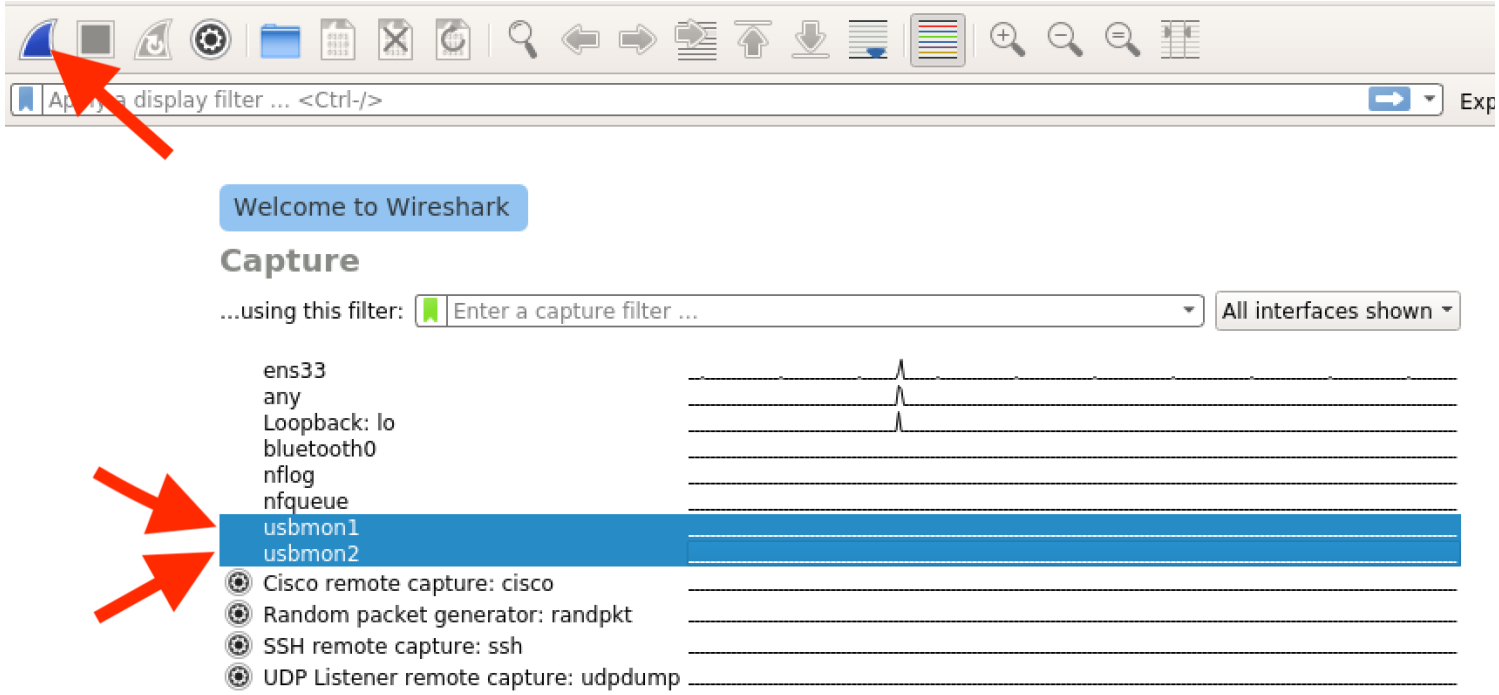
Let’s see if Predator Sense uses USB!

But how can I check that?



WIRESHARK

The world's foremost and widely-used network protocol analyzer



# Let’s dive deeper

- Monitor hard drive activities
- Monitor Windows registry changes
- Check files inside Predator Sense
- Check startup service for Predator Sense
- Reverse engineering?

# De-compiling Predator Sense

# Source Code

JetBrains dotPeek

File View Navigate Inspect Tools Windows Help

Assembly Explorer

Type to search

- PredatorSense (3.0.3146.0, x64, .Net Framework v4.5)
  - Metadata
  - References
  - Resources
  - Generic.Core.WebContext
  - IWshRuntimeLibrary
  - Library
  - PredatorSense
    - AddCenterPage
    - Advanced_Setting_CheckBox
    - Advanced_Setting_Dropbox_GPUMode
    - Advanced_Setting_RadioButton
    - App
    - AppCenter_AddWindow
    - AppCenter_AppExiting
    - AppCenter_DeleteShortcut
    - AppItem
    - AppManager_Uninstall_Application
    - AppManger_List_Item
    - Attention
    - BIOSLogoAnimation
    - BIOSLogoSetting
    - Clear_history
    - ColorToBrushConverter
    - CommonFunction
    - dataItem
    - FanControlPage
    - FanControlPage_Ghibli

datorSense.cs | PsLightingKeyboardUI.cs | LightingDynamicUI.cs | LightingDynamicUI_Covini.cs | Com

```
19 using System.Linq;
20 using System.Management;
21 using System.Runtime.CompilerServices;
22 using System.Runtime.InteropServices;
23 using System.Security;
24 using System.Text;
25 using System.Text.RegularExpressions;
26 using System.Threading.Tasks;
27 using System.Windows;
28 using System.Windows.Media;
29 using System.Windows.Media.Imaging;
30 using TsDotNetLib;
31 using Windows.Networking.Connectivity;
32
33 namespace PredatorSense
34 {
35   public class CommonFunction
36   {
37     public static int Last_CPU_Angle = 0;
38     public static int Current_CPU_Angle = 0;
39     public static int Current_CPU_MaxMHz = 0;
40     public static int Current_CPU_MaxAngle = 0;
41     public static int CPUNormal_MaxMHz = 0;
42     public static int CPUFaster_MaxMHz = 0;
43     public static int CPUBoosted_MaxMHz = 0;
44     public static int CPUTurbo_MaxMHz = 0;
45     public static int[] Last_GPU_Angle = new int[4];
46     public static int[] Current_GPU_Angle = new int[4];
47     public static int[] Current_GPU_MaxMHz = new int[4];
48     public static int[] Current_GPU_MaxAngle = new int[4];
49     public static int[] GPUNormal_MaxMHz = new int[4];
50     public static int[] GPUFaster_MaxMHz = new int[4];
51     public static int[] GPUBoosted_MaxMHz = new int[4];
52     public static int[] GPUTurbo_MaxMHz = new int[4];
53     public static string SESSION_COMMONPARAMETER = "CommonPatameter";
54     public static string SESSION_GAMEPATH = "GamePath";
55     public static string SESSION_HOTKEY_PROFILE = "HotkeyProfile";
56     public static string SESSION_LIGHTING_PROFILE = "LightingProfile";
```

Quick Find

Find what:

Look in: Document

Find options

Find Next

Process Explorer

Find Results

**No search results available**

Use Shift+F12

# Searching for RGB

## In Predator Sense Source Code

Assembly Explorer

rgb

- mscorlib (4.0.0.0, x64)
  - System.Diagnostics
    - StackFrameHelper
      - rgMethodBase:MethodBase[]
  - System.Reflection
    - RuntimeAssembly
      - GetRawBytes():byte[]
      - GetRawBytes(RuntimeAssembly assembly, ObjectHandleOnStac
  - System.Security.Cryptography
    - CryptoAPITransform
      - _rgbKey:byte[]
    - Utils
      - SetKeyParamRgb(SafeKeyHandle hKey, int param, byte[] value,
- PredatorSense (3.0.3146.0, x64, .Net Framework v4.5)
  - Library
    - IMAGELISTDRAWPARAMS
      - rgbBk:int
      - rgbFg:int
  - PredatorSense
    - LightingPage
      - Change_Zone_Brightness_RGB(int brightness):void
    - LightingPage_Covini
      - Change_Zone_Brightness_RGB(int brightness):void
    - LightingPage_Ghibli
      - Change_Zone_Brightness_RGB(int brightness):void
    - MacroSettingPage
      - right_group_TextBlock:TextBlock
      - r_group_name_TextBox:TextBox

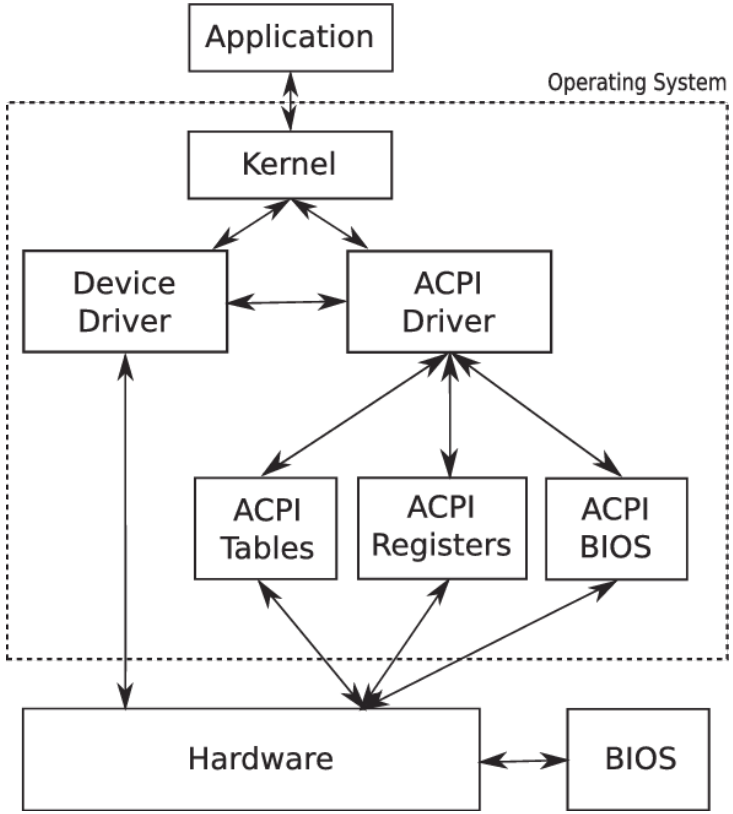
PredatorSense.cs | LightingPage.cs

```
704         case 0:
705           num2 = 0.0;
706           break;
707         case 50:
708           num2 = 0.5;
709           break;
710         case 100:
711           num2 = 1.0;
712           break;
713         default:
714           num2 = 1.0;
715           break;
716       }
717       try
718       {
719         LightingProfileXML lightingProfileXml = new LightingProfileXML(CommonFunction.lighting_profile_path + this.current_lighting_prof
720         for (int index = 0; index < ((IEnumerable<string>) this.ZoneName).Count<string>(); ++index)
721         {
722           Color color = (Color) ColorConverter.ConvertFromString(lightingProfileXml.lightingeffects_group_content.lightingeffects_conten
723           ulong uint64_1 = Convert.ToUInt64(Math.Floor(Convert.ToDouble((object) color.R, (IFormatProvider) CultureInfo.InvariantCulture
724           ulong uint64_2 = Convert.ToUInt64(Math.Floor(Convert.ToDouble((object) color.G, (IFormatProvider) CultureInfo.InvariantCulture
725           ulong uint64_3 = Convert.ToUInt64(Math.Floor(Convert.ToDouble((object) color.B, (IFormatProvider) CultureInfo.InvariantCulture
726           if (index + 1 == 1)
727             intput = (ulong) (1L | (long) uint64_1 << 8 | (long) uint64_2 << 16 | (long) uint64_3 << 24);
728           else if (index + 1 == 2)
729             intput = (ulong) (2L | (long) uint64_1 << 8 | (long) uint64_2 << 16 | (long) uint64_3 << 24);
730           else if (index + 1 == 3)
731             intput = (ulong) (4L | (long) uint64_1 << 8 | (long) uint64_2 << 16 | (long) uint64_3 << 24);
732           else if (index + 1 == 4)
733             intput = (ulong) (8L | (long) uint64_1 << 8 | (long) uint64_2 << 16 | (long) uint64_3 << 24);
734           WMIFunction.WMISetGamingRgbKbSetting(intput).GetAwaiter();
735         }
736       }
737       catch (Exception ex)
738       {
739         TsDotNetLib.Log.LogWrite(this._log, LogType.Exception, MethodBase.GetCurrentMethod().Name, "ex = " + ex.ToString());
740       }
741     }
```

# WMI & ACPI

Windows Management Instrumentation(WMI)
Advanced Configuration and Power Interface(ACPI)

WMI Is a subset of ACPI which allows interacting with low-level hardware features of a system.



# WMI Explorer

## A mini tool to view\interact with WMI Methods

Classes (1550) | Search

Quick Filter:

Classes

| Name | Lazy... | Description | Path |
|---|---|---|---|
| AcerBiosConfiguration... | False | Acer BIOS Setting To... | \\DESKTOP-UDA2M... |
| AcerGamingFunction | False | Class used to Gamin... | \\DESKTOP-UDA2M... |
| AcpiControlStatus | False | ACPI control and status | \\DESKTOP-UDA2M... |
| AcpiGenAddr | False | Generic Address Des... | \\DESKTOP-UDA2M... |
| AcpiPct | False | Acpi _PCT | \\DESKTOP-UDA2M... |
| AcpiPss | False | Processor Performan... | \\DESKTOP-UDA2M... |
| AcpiPssState | False | Acpi _PSS State | \\DESKTOP-UDA2M... |
| ACPITrace | False | ACPI Driver Trace Pr... | \\DESKTOP-UDA2M... |
| ActivityTransfer | False | Activity Transfer Event | \\DESKTOP-UDA2M... |
| AllocateSegment | False | Segment allocation | \\DESKTOP-UDA2M... |
| AllocationTick | False | Allocation Tick | \\DESKTOP-UDA2M... |
| ALPC | False | ALPC Event | \\DESKTOP-UDA2M... |
| ALPC_Receive_Mess... | False | ALPC Receive Mess... | \\DESKTOP-UDA2M... |
| ALPC_Send_Message | False | ALPC Send Message | \\DESKTOP-UDA2M... |
| ALPC_Unwait | False | ALPC Unwait | \\DESKTOP-UDA2M... |
| ALPC_Wait_For_New... | False | ALPC Wait For New ... | \\DESKTOP-UDA2M... |
| ALPC_Wait_For_Reply | False | ALPC Wait For Reply | \\DESKTOP-UDA2M... |
| AMLIEvalData1 | False | AMLI Eval Trace Pro... | \\DESKTOP-UDA2M... |
| AMLIEvalData1_Type... | False | AMLI Eval Trace Pro... | \\DESKTOP-UDA2M... |
| AntiStarvationBoost | False | Anti Starvation Boost ... | \\DESKTOP-UDA2M... |
| APGeAction | False | APGe WMI Method, V... | \\DESKTOP-UDA2M... |
| APGeEvent | False | APGe WMI Event, Ve... | \\DESKTOP-UDA2M... |
| AtaportGuid | False | ATA Port Driver Traci... | \\DESKTOP-UDA2M... |
| AuthenticodeVerificati... | False | AuthenticodeVerificati... | \\DESKTOP-UDA2M... |
| AUTHFWCFG | False | AuthFw NetShell Plugin | \\DESKTOP-UDA2M... |
| AutoBoostClearFloor | False | AutoBoost Clear Floo... | \\DESKTOP-UDA2M... |
| AutoBoostEntryExhau... | False | AutoBoost Entry Exha... | \\DESKTOP-UDA2M... |
| AutoBoostSetFloor | False | AutoBoost Set Floor ... | \\DESKTOP-UDA2M... |
| BatteryControl | False | Class used to control ... | \\DESKTOP-UDA2M... |
| BatteryCycleCount | False | Battery Cycle Count | \\DESKTOP-UDA2M... |
| BatteryFullChargedCa... | False | Battery Fully Charged... | \\DESKTOP-UDA2M... |
| BatteryRuntime | False | Estimated Runtime | \\DESKTOP-UDA2M... |
| BatteryStaticData | False | Battery Static Data | \\DESKTOP-UDA2M... |

Instances (1) | Properties (2) | Methods (25) | Query | Script | Logging

Class Properties

| Property Name | Type | Enumeration Available | Lazy | Description |
|---|---|---|---|---|
| Active | Boolean | False | False | |
| InstanceName | String | False | False | |

**AcerGamingFunction Class**

Class used to Gaming Function, Version 2.76

**Class Qualifiers:**

Description
dynamic - True
guid - {7A4DDFE7-5B5D-40B4-8595-4408E0CC7F56}
Locale - MS\0x409
provider - WmiProv
WMI - True

**Properties:**

*Active - Boolean*

Qualifiers: CIMTYPE, read

[Description Not Available]

# Playing with RGB Colors

## And check WMI changes

WMI Explorer 2.0.0.2 (Administrator)

File Launch Help

Computer: DESKTOP-UDA2M7U Connect

Mode: Asynchronous / Synchronous

Class Enumeration Options — Filter: % — Include System Classes, Include Perf Classes, Include CIM Classes, Include MSFT Classes, Refresh Classes

Namespaces:
- \\DESKTOP-UDA2M7U
  - ROOT\Appv
  - ROOT\CIMV2
  - ROOT\Cli
  - ROOT\DEFAULT
  - ROOT\directory
  - ROOT\Hardware
  - ROOT\Interop
  - ROOT\Microsoft
  - ROOT\msdtc
  - ROOT\PEH
  - ROOT\Policy
  - ROOT\RSOP
  - ROOT\SECURITY
  - ROOT\SecurityCenter
  - ROOT\SecurityCenter2
  - ROOT\ServiceModel
  - ROOT\StandardCimv2
  - ROOT\subscription
  - ROOT\WMI
    - ROOT\WMI\ms_409

Classes (1550) | Search

Quick Filter:

Classes

| Name | Lazy... | Description | Path |
|---|---|---|---|
| AcerBiosConfiguration... | False | Acer BIOS Setting To... | \\DESKTOP-UDA2M... |
| AcerGamingFunction | False | Class used to Gamin... | \\DESKTOP-UDA2M... |
| AcpiControlStatus | False | ACPI control and status | \\DESKTOP-UDA2M... |
| AcpiGenAddr | False | Generic Address Des... | \\DESKTOP-UDA2M... |
| AcpiPct | False | Acpi _PCT | \\DESKTOP-UDA2M... |
| AcpiPss | False | Processor Performan... | \\DESKTOP-UDA2M... |
| AcpiPssState | False | Acpi _PSS State | \\DESKTOP-UDA2M... |
| ACPITrace | False | ACPI Driver Trace Pr... | \\DESKTOP-UDA2M... |
| ActivityTransfer | False | Activity Transfer Event | \\DESKTOP-UDA2M... |
| AllocateSegment | False | Segment allocation | \\DESKTOP-UDA2M... |
| AllocationTick | False | Allocation Tick | \\DESKTOP-UDA2M... |
| ALPC | False | ALPC Event | \\DESKTOP-UDA2M... |
| ALPC_Receive_Mess... | False | ALPC Receive Mess... | \\DESKTOP-UDA2M... |
| ALPC_Send_Message | False | ALPC Send Message | \\DESKTOP-UDA2M... |
| ALPC_Unwait | False | ALPC Unwait | \\DESKTOP-UDA2M... |
| ALPC_Wait_For_New... | False | ALPC Wait For New ... | \\DESKTOP-UDA2M... |
| ALPC_Wait_For_Reply | False | ALPC Wait For Reply | \\DESKTOP-UDA2M... |
| AMLIEvalData1 | False | AMLI Eval Trace Pro... | \\DESKTOP-UDA2M... |
| AMLIEvalData1_Type... | False | AMLI Eval Trace Pro... | \\DESKTOP-UDA2M... |
| AntiStarvationBoost | False | Anti Starvation Boost ... | \\DESKTOP-UDA2M... |
| APGeAction | False | APGe WMI Method, V... | \\DESKTOP-UDA2M... |
| APGeEvent | False | APGe WMI Event, Ve... | \\DESKTOP-UDA2M... |
| AtaportGuid | False | ATA Port Driver Traci... | \\DESKTOP-UDA2M... |
| AuthenticodeVerificati... | False | AuthenticodeVerificati... | \\DESKTOP-UDA2M... |
| AUTHFWCFG | False | AuthFw NetShell Plugin | \\DESKTOP-UDA2M... |
| AutoBoostClearFloor | False | AutoBoost Clear Floo... | \\DESKTOP-UDA2M... |
| AutoBoostEntryExhau... | False | AutoBoost Entry Exha... | \\DESKTOP-UDA2M... |
| AutoBoostSetFloor | False | AutoBoost Set Floor ... | \\DESKTOP-UDA2M... |
| BatteryControl | False | Class used to control ... | \\DESKTOP-UDA2M... |
| BatteryCycleCount | False | Battery Cycle Count | \\DESKTOP-UDA2M... |
| BatteryFullChargedCa... | False | Battery Fully Charged... | \\DESKTOP-UDA2M... |
| BatteryRuntime | False | Estimated Runtime | \\DESKTOP-UDA2M... |
| BatteryStaticData | False | Battery Static Data | \\DESKTOP-UDA2M... |

Instances (1) | Properties (2) | Methods (25) | Query | Script | Logging

Instance Options — Quick Filter: — Show Null Values, Show System Properties — Refresh Instances, Refresh Object

Instances: AcerGamingFunction.InstanceNam

- Show MOF
- Copy MOF
- Copy Path
- Copy Relative Path
- Execute Method
  - GetCPUOverclockingProfile
  - GetGamingFanBehavior
  - GetGamingFanSpeed
  - GetGamingFanTable
  - GetGamingKBBacklight
  - GetGamingLED
  - GetGamingLEDBehavior
  - GetGamingLEDColor
  - GetGamingMiscSetting
  - GetGamingProfile
  - GetGamingProfileSetting
  - GetGamingRgbKb
  - GetGamingSysInfo
  - SetCPUOverclockingProfile
  - SetGamingFanBehavior
  - SetGamingFanSpeed
  - SetGamingFanTable
  - SetGamingKBBacklight
  - SetGamingLED
  - SetGamingLEDBehavior
  - SetGamingLEDColor
  - SetGamingMiscSetting
  - SetGamingProfile
  - SetGamingProfileSetting
  - SetGamingRgbKb

Properties

| | |
|---|---|
| *InstanceName | ACPI\PNP0C14\APGe_0 |
| Active | True |

WQL Query (Selected Object)

Query: SELECT * FROM AcerGamingFunction WHERE InstanceName='ACPI\\PNP0C14\\APGe_0' — Execute

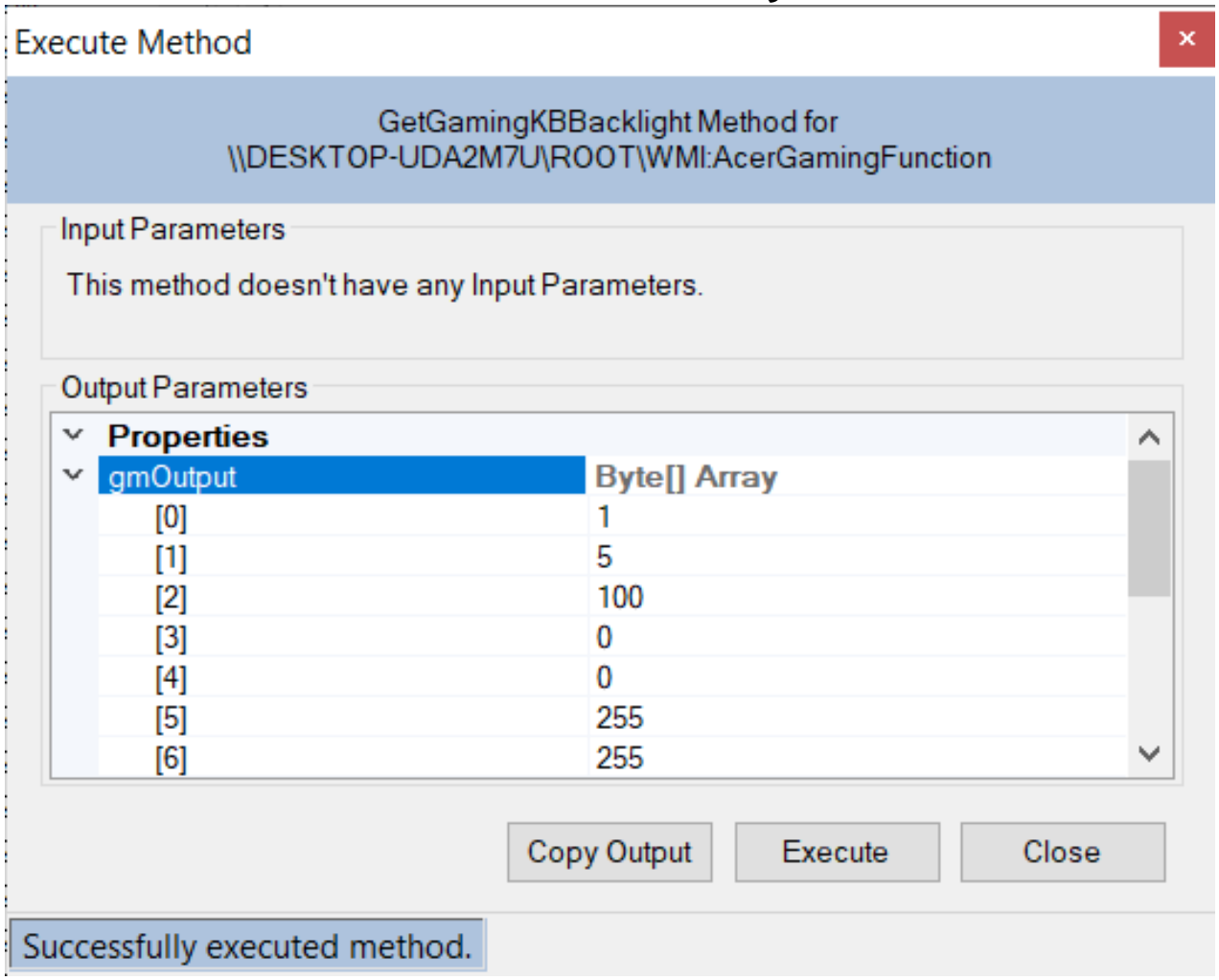
Activate Windows
Go to Settings to activate Windows.

Showing 1550 cached classes for ROOT\WMI from 11:32:16 | Retrieved 1 instances from AcerGamingFunction

Time to Enumerate Instances: 00:00.003

# Playing with RGB Colors

And check WMI changes



# What is the meaning of these bytes?!

Change Keyboard Animation Direction:
From **Left-To-Right** to **Right-To-Left**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Before** | 3 | 6 | 100 | 8 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **After** | 3 | 6 | 100 | 8 | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

# What is the meaning of these bytes?!

Change Keyboard Animation Color:
From **Blue** to **White**

Blue RGB color: 0, 0, 255

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Before** | 1 | 5 | 100 | 0 | 0 | **0** | **0** | **255** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **After** | 1 | 5 | 100 | 0 | 0 | **255** | **255** | **255** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

White RGB color: 255, 255, 255

# Let’s wrap it up

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Action | Effect Mode | Speed | Brightness | Unknown | Animation Direction | RGB Red | RGB Green | RGB Blue |

# Implementing the solution

1) How can I interact with WMI functions in Linux?
2) How can I implement it in C programming language?
3) Can I simply use GUID and method name in Linux?
4) Where should I start?

# Faustus

Experimental unofficial Linux platform driver module for **ASUS TUF Gaming series** laptops

Looks exactly like what I'm trying to do, but for Asus ( my laptop is Acer)

- What did he change?!
- He modified a file related to Asus WMI in Linux kernel
- For some reason, WMI Method names are not available in Linux! I'll have to look for equivalent Method IDs

# What are the changes in Faustus?

Let's detect Faustus  changes with actual kernel code for Asus-wmi driver in Linux Kernel
( linux/drivers/platform/x86/asus-wmi.c )

kernel.c (/home/black-hole/projects/devops/backend/phoenix/TESTME)

```
#include <linux/pci.h>
#include <linux/pci_hotplug.h>
#include <linux/hwmon.h>
#include <linux/hwmon-sysfs.h>
#include <linux/debugfs.h>
#include <linux/seq_file.h>
#include <linux/platform_data/x86/asus-wmi.h>
#include <linux/platform_device.h>
#include <linux/thermal.h>
#include <linux/acpi.h>
#include <linux/dmi.h>
#include <acpi/video.h>

#include "asus-wmi.h"

MODULE_AUTHOR("Corentin Chary <corentin.chary@gmail.com>, "
              "Yong Wang <yong.y.wang@intel.com>");
MODULE_DESCRIPTION("Asus Generic WMI Driver");
MODULE_LICENSE("GPL");

#define to_asus_wmi_driver(pdrv)                        \
    (container_of((pdrv), struct asus_wmi_driver, platform_driver))

#define ASUS_WMI_MGMT_GUID  "97845ED0-4E6D-11DE-8A39-0800200C9A66"

#define NOTIFY_BRNUP_MIN        0x11
#define NOTIFY_BRNUP_MAX        0x1f
#define NOTIFY_BRNDOWN_MIN      0x20
#define NOTIFY_BRNDOWN_MAX      0x2e
#define NOTIFY_KBD_BRTUP        0xc4
#define NOTIFY_KBD_BRTDWN       0xc5
#define NOTIFY_KBD_BRTTOGGLE        0xc7

#define ASUS_FAN_DESC           "cpu_fan"
#define ASUS_FAN_MFUN           0x13
#define ASUS_FAN_SFUN_READ      0x06
#define ASUS_FAN_SFUN_WRITE     0x07
#define ASUS_FAN_CTRL_MANUAL        1
```

faustus.c (/home/black-hole/projects/devops/backend/phoenix/TESTME)

```
/* WMI Methods */
#define ASUS_WMI_METHODID_SPEC          0x43455053 /* BIOS SPECification */
#define ASUS_WMI_METHODID_SFBD      0x44424653 /* Set First Boot Device */
#define ASUS_WMI_METHODID_GLCD      0x44434C47 /* Get LCD status */
#define ASUS_WMI_METHODID_GPID      0x44495047 /* Get Panel ID?? (Resol) */
#define ASUS_WMI_METHODID_QMOD      0x444F4D51 /* Quiet MODe */
#define ASUS_WMI_METHODID_SPLV      0x4C425053 /* Set Panel Light Value */
#define ASUS_WMI_METHODID_AGFN      0x4E464741 /* FaN? */
#define ASUS_WMI_METHODID_SFUN      0x4E554653 /* FUNCtionalities */
#define ASUS_WMI_METHODID_SDSP      0x50534453 /* Set DiSPlay output */
#define ASUS_WMI_METHODID_GDSP      0x50534447 /* Get DiSPlay output */
#define ASUS_WMI_METHODID_DEVP      0x50564544 /* DEVice Policy */
#define ASUS_WMI_METHODID_OSVR      0x5256534F /* OS VeRsion */
#define ASUS_WMI_METHODID_DCTS      0x53544344 /* Device status (DCTS) */
#define ASUS_WMI_METHODID_DSTS      0x53545344 /* Device status (DSTS) */
#define ASUS_WMI_METHODID_BSTS      0x53545342 /* Bios STatuS ? */
#define ASUS_WMI_METHODID_DEVS      0x53564544 /* DEVice Set */
#define ASUS_WMI_METHODID_CFVS      0x53564643 /* CPU Frequency Volt Set */
#define ASUS_WMI_METHODID_KBFT      0x5446424B /* KeyBoard FilTer */
#define ASUS_WMI_METHODID_INIT      0x54494E49 /* INITialize */
#define ASUS_WMI_METHODID_HKEY      0x59454B48 /* Hot KEY ?? */

#define ASUS_WMI_UNSUPPORTED_METHOD 0xFFFFFFFE

/* Wireless */
#define ASUS_WMI_DEVID_HW_SWITCH    0x00010001
#define ASUS_WMI_DEVID_WIRELESS_LED 0x00010002
#define ASUS_WMI_DEVID_CWAP     0x00010003
#define ASUS_WMI_DEVID_WLAN     0x00010011
#define ASUS_WMI_DEVID_WLAN_LED     0x00010012
#define ASUS_WMI_DEVID_BLUETOOTH    0x00010013
#define ASUS_WMI_DEVID_GPS      0x00010015
#define ASUS_WMI_DEVID_WIMAX        0x00010017
#define ASUS_WMI_DEVID_WWAN3G       0x00010019
#define ASUS_WMI_DEVID_UWB      0x00010021

/* Leds */
/* 0x000200XX and 0x000400XX */
#define ASUS_WMI_DEVID_LED1     0x00020011
```

# Faustus

Let’s ask the creator of Faustus how he did find the method IDs

**JafarAkhondali** commented on Mar 5

Hi,
May I know how did you find these device addresses?

```
#define ASUS_WMI_DEVID_KBD_RGB          0x00100056
#define ASUS_WMI_DEVID_KBD_RGB2         0x00100057
```

**hackbnw** commented on Mar 7 Owner

Hi, trial and error mostly. It can be guessed by examining DSDT very carefully, as they are taking a very specific pattern of arguments consistent with RGB.

1

# Implementing a WMI Driver in Linux

- https://lwn.net/Articles/367630/ (2009)
- https://lwn.net/Articles/391230/ (2010)
- https://wiki.ubuntu.com/Kernel/Reference/WMI
- I strongly suggest to read above resources yourself, cause they are not suitable for this presentation
- TLDR; we'll need a table called "Discrete System Descriptor Table" (DSDT)
- This table is available on /sys/firmware/acpi/tables/DSDT
- After reading above resources several times, I realized there is an interesting section in DSDT called Managed Object Format (MOF)

# Managed Object Format (MOF)

## WQxx buffers

A lot of WMI implementations contain a WQxx declaration in the DSDT.

The WQxx buffers are declared as follows:

```
          Name (WQM0, Buffer (0x047A)
           {
               /* 0000 */    0x46, 0x4F, 0x4D, 0x42, 0x01, 0x00, 0x00, 0x00,
               /* 0008 */    0x6A, 0x04, 0x00, 0x00, 0xD8, 0x11, 0x00, 0x00,
...
           }
```

These are generally large blobs of Managed Object Format (MOF) binary data embedded in the AML. Windows will evaluate the WQxx() buffer which returns to Windows the MOF binary which describes all data blocks, WMI methods, and events for the device in a compressed binary format. As yet, there don't seem to be any MOF decompilers that can reverse engineer these blobs back into the MOF language, so we are at the mercy of getting descriptions of the MOF from the BIOS vendor to figure out how this maps to the WMI AML methods described inside the MOF.

# Decompiling MOF file

Luckily, someone commented a solution to de-compile MOF format on https://lwn.net.

It’s possible to use a CLI tool (wmiofck.exe) in Windows Driver Development Kit to decompile MOF data.

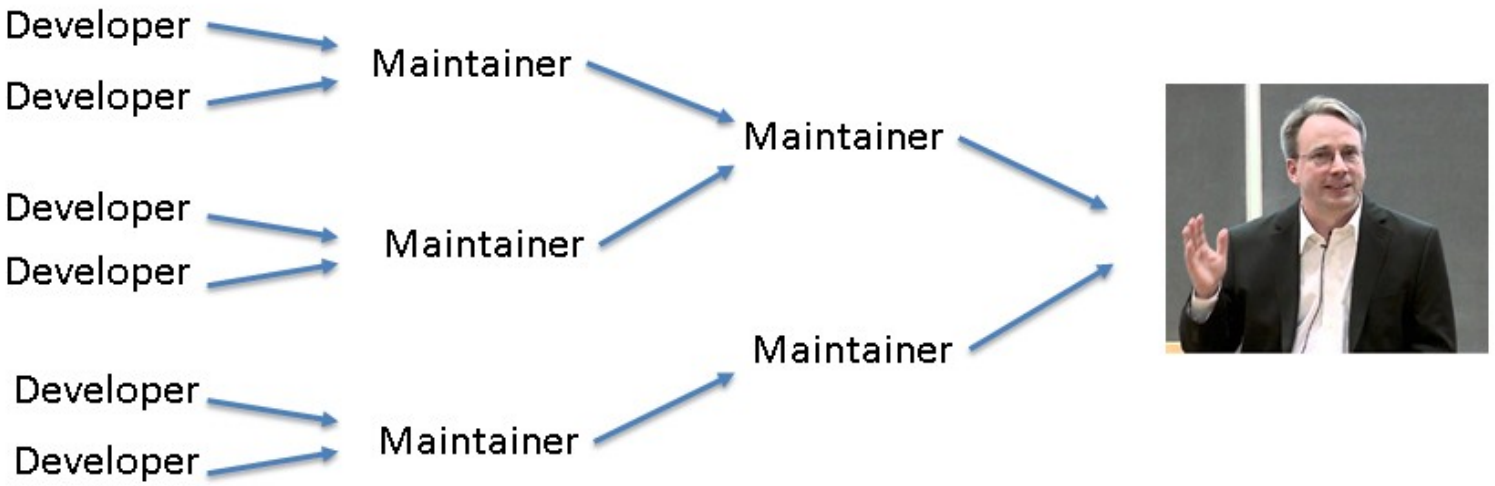
And now we can detect related Method ID of a method name!

```
//...
#define SetGamingKBBacklight     20
typedef struct _SetGamingKBBacklight_IN
{
    UCHAR gmInput[16];
    #define SetGamingKBBacklight_IN_gmInput_SIZE sizeof(UCHAR[16])
    #define SetGamingKBBacklight_IN_gmInput_ID 1

} SetGamingKBBacklight_IN, *PSetGamingKBBacklight_IN;
//...
```

# Adding the feature in Linux kernel

# Adding the feature in Linux kernel



# My first contribution in Linux kernel

1) Register events when Acer gaming functions GUID is found
2) Communicate with user-space when data is passed to driver using a character device
3) Pass data to specific WMI method
4) Clone Linux kernel source code and create a patch file
5) Check & apply Linux kernel contribution guide line
   https://www.kernel.org/doc/html/latest/process/submitting-patches.html
6) Find required reviewer(s) related to your patch
7) Join the mailing list related to your changes
8) Send patch file using “git send-email” to not mess email patch format
9) Wait for kernel maintainers to review your code
10) Answer to maintainers comments
11) Fix comments, send new version of patch as new version, Repeat until it’s accepted
12) My first patch link: https://www.spinics.net/lists/platform-driver-x86/msg25742.html
13) And surprise! My patch got rejected :))

# Why did my first patch got rejected?

- Linux kernel already have a sub-module for controlling LEDs, but it misses some features for my case
- I wasn't able to merge this patch into Linux kernel, so instead I did almost same steps for Implementing Turbo mode feature, and was able to ship it to Linux Kernel
- Final patch for turbo mode (that got accepted):

  https://www.spinics.net/lists/platform-driver-x86/msg27223.html
- As many people need these features, I've made a project on Github to install it as a kernel module:

  https://github.com/JafarAkhondali/acer-helios-300-rgb-keyboard-linux-module
- Some users tried this on their laptops and surprisingly the project worked on other Acer Predator series (such as Nitro and Triton)
- Someone started implementing a GUI for this tool

My favorite phrases from

# Linux Kernel Coding Style

#سُسس - ماست
احاديث لينوس توروالدز(ع)

- “First off, I’d suggest printing out a copy of the GNU coding standards, and NOT read it. Burn them, it’s a great symbolic gesture.”
- Indentation: “Tabs are 8 characters, and thus indentations are also 8 characters. There are heretic movements that try to make indentations 4 (or even 2!) characters deep, and that is akin to trying to define the value of PI to be 3.”

  Now, some people will claim that having 8-character indentations makes the code move too far to the right, and makes it hard to read on a 80-character terminal screen. The answer to that is that if you need more than 3 levels of indentation, you’re screwed anyway, and should fix your program.
- Don’t put multiple assignments on a single line. Kernel coding style is super simple. Avoid tricky expressions.

My favorite phrases from

# Linux Kernel Coding Style

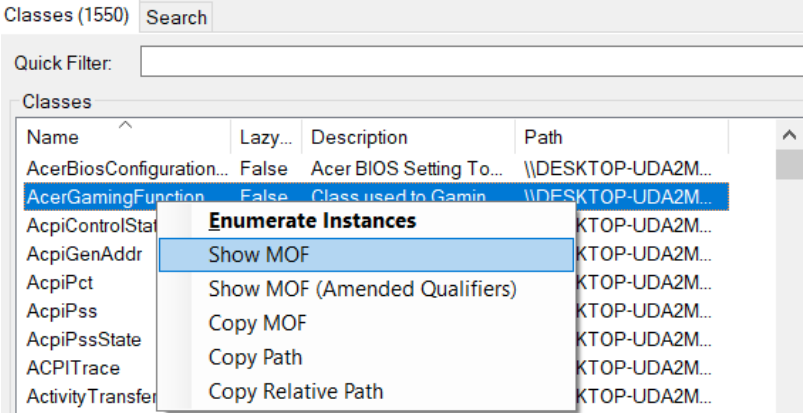
#سُسس - ماست
احاديث لينوس توروالدز(ع)

- Encoding the type of a function into the name (so-called Hungarian notation) is brain damaged - the compiler knows the types anyway and can check those, and it only confuses the programmer. No wonder MicroSoft makes buggy programs.

- Comments are good, but there is also a danger of over-commenting. NEVER try to explain HOW your code works in a comment: it's much better to write the code so that the working is obvious, and it's a waste of time to explain badly written code.

- Generally, you want your comments to tell WHAT your code does, not HOW.

# The thing that I wish I knew sooner



# The thing that I wish I knew sooner

MOF

```
[WMI, dynamic: ToInstance, provider("WmiProv"), Locale("MS\\0x409"), Description("Class used to Gaming Function, Version 2.76"), guid("{7A4DDFE7-5B5D-40B4-8595-4408E0CC7F56}")]
class AcerGamingFunction
{
	[key, read] string InstanceName;
	[read] boolean Active;
	[WmiMethodId(1), Implemented, read, write, Description("Set Acer Gaming Profile Configuration.")] void SetGamingProfile([in] uint64 gmInput, [out] uint32 gmOutput);
	[WmiMethodId(2), Implemented, read, write, Description("Set Acer Gaming LED Behavior.")] void SetGamingLED([in] uint64 gmInput, [out] uint32 gmOutput);
	[WmiMethodId(3), Implemented, read, write, Description("Get Acer Gaming Profile Configuration.")] void GetGamingProfile([in] uint32 gmInput, [out] uint64 gmOutput);
	[WmiMethodId(4), Implemented, read, write, Description("Get Acer Gaming LED Behavior.")] void GetGamingLED([in] uint32 gmInput, [out] uint64 gmOutput);
	[WmiMethodId(5), Implemented, read, write, Description("Get Acer Gaming System Information.")] void GetGamingSysInfo([in] uint32 gmInput, [out] uint64 gmOutput);
	[WmiMethodId(6), Implemented, read, write, Description("Set Acer Gaming RGB Keyboard Setting.")] void SetGamingRgbKb([in] uint32 gmInput, [out] uint64 gmOutput);
	[WmiMethodId(7), Implemented, read, write, Description("Get Acer Gaming RGB Keyboard Setting.")] void GetGamingRgbKb([in] uint32 gmInput, [out] uint64 gmOutput);
	[WmiMethodId(8), Implemented, read, write, Description("Set Acer Gaming Profile Setting.")] void SetGamingProfileSetting([in] uint64 gmInput, [out] uint32 gmOutput);
	[WmiMethodId(9), Implemented, read, write, Description("Get Acer Gaming Profile Setting.")] void GetGamingProfileSetting([in] uint32 gmInput, [out] uint64 gmOutput);
	[WmiMethodId(10), Implemented, read, write, Description("Set Acer Gaming LED Group Behavior.")] void SetGamingLEDBehavior([in] uint64 gmInput, [out] uint32 gmOutput);
	[WmiMethodId(11), Implemented, read, write, Description("Get Acer Gaming LED Group Behavior.")] void GetGamingLEDBehavior([in] uint32 gmInput, [out] uint64 gmOutput);
	[WmiMethodId(12), Implemented, read, write, Description("Set Acer Gaming LED Group Color.")] void SetGamingLEDColor([in] uint64 gmInput, [out] uint32 gmOutput);
	[WmiMethodId(13), Implemented, read, write, Description("Get Acer Gaming LED Group Color.")] void GetGamingLEDColor([in] uint32 gmInput, [out] uint64 gmOutput);
	[WmiMethodId(14), Implemented, read, write, Description("Set Acer Gaming Fan Group Behavior.")] void SetGamingFanBehavior([in] uint64 gmInput, [out] uint32 gmOutput);
	[WmiMethodId(15), Implemented, read, write, Description("Get Acer Gaming Fan Group Behavior.")] void GetGamingFanBehavior([in] uint32 gmInput, [out] uint64 gmOutput);
	[WmiMethodId(16), Implemented, read, write, Description("Set Acer Gaming Fan Group Speed.")] void SetGamingFanSpeed([in] uint64 gmInput, [out] uint32 gmOutput);
	[WmiMethodId(17), Implemented, read, write, Description("Get Acer Gaming Fan Group Speed.")] void GetGamingFanSpeed([in] uint32 gmInput, [out] uint64 gmOutput);
	[WmiMethodId(18), Implemented, read, write, Description("Set Acer Gaming Fan Table.")] void SetGamingFanTable([in] uint64 gmInput, [out] uint32 gmOutput);
	[WmiMethodId(19), Implemented, read, write, Description("Get Acer Gaming Fan Table.")] void GetGamingFanTable([out] uint64 gmOutput);
	[WmiMethodId(20), Implemented, read, write, Description("Set Acer Gaming Keyboard Backlight Behavior.")] void SetGamingKBBacklight([in, MAX(16)] uint8 gmInput[], [out] uint32 gmOutput);
	[WmiMethodId(21), Implemented, read, write, Description("Get Acer Gaming Keyboard Backlight Behavior.")] void GetGamingKBBacklight([out] uint8 gmReturn, [out, MAX(15)] uint8 gmOutput[]);
	[WmiMethodId(22), Implemented, read, write, Description("Set Acer Gaming Miscellaneous Setting.")] void SetGamingMiscSetting([in] uint64 gmInput, [out] uint32 gmOutput);
	[WmiMethodId(23), Implemented, read, write, Description("Get Acer Gaming Miscellaneous Setting.")] void GetGamingMiscSetting([in] uint32 gmInput, [out] uint64 gmOutput);
	[WmiMethodId(24), Implemented, read, write, Description("Set CPU Overclocking Profile.")] void SetCPUOverclockingProfile([in] uint8 OCProfile, [in, MAX(512)] uint8 OCStructure[], [out] uint8 ReturnCode, [out, MAX(3)] uint8 Reserved[]);
	[WmiMethodId(25), Implemented, read, write, Description("Get CPU Overclocking Profile.")] void GetCPUOverclockingProfile([in, MAX(4)] uint8 Reserved[], [out] uint8 ReturnCode, [out] uint8 ReturnOCProfile, [out, MAX(512)] uint8 OCStructure[]);
};
```

# Thanks for your time!

- Questions?

- Email: Jafar.akhoondali@gmail.com
- Telegram: @deallocate
- Twitter: @theDeallocated
- Github: https://github.com/JafarAkhondali/
- Other ways:
  https://akhondali.ir/