UEFI Development Exploration 70- YIE001PCIe Development Board (06 UEFI Driver)

原创 luobing4365 Posted on 2021-01-17 13:27:40 Views:706 Collection 2 Likes copyright

Category Column: UEFI Development Article Tags: UEFI BIOS Low-level application development UEFI Drivers EDK2

UEFI Development This column includes this content 503 Subscribe 104 articles Subscribe to our column

(Please keep it-> Author: Luo Bing https://blog.csdn.net/luobing4365)

In the previous two articles, we learned about the basic architecture of the UEFI driver model . Based on this, this article will use an interesting UEFI driver GopRotate and a self-written test UEFI application TestGopRotate to demonstrate the operation and testing process of the UEFI driver.

1 UEFI driver GopRotate

This example is provided by apop2, and the source code repository is https://github.com/apop2/GopRotate . The GopRotate project contains 5 source files: ComponentName.c, GopRotate.c, GopRotate.h, GopRotateBlt.c, and GopRotate.inf. The functions of these files are as follows:

1) ComponentName.c. Implements the interface function of EFI_COMPONENT_NAME_PROTOCOL and provides the name of the UEFI driver;
2) GopRotate.c. Implements EFI_DRIVER_BINDING_PROTOCOL and its interface functions, installs EFI_DRIVER_BINDING_PROTOCOL and EFI_COMPONENT_NAME_PROTOCOL, and implements the Protocol for the UEFI Shell interface rotation function;
3) GopRotate.h. Defines the data structure required by the driver and the prototype of the Protocol interface function;
4) GopRotateBlt.c. Implements the function of UEFI Shell interface rotation and the implementation of the Protocol interface function that provides this function;
5) GopRotate.inf, compiles the INF file of the UEFI driver.

To realize the rotation of the UEFI Shell interface, the sample project GopRotate mainly does the following work:

- Find the controller with EFI_GRAPHICS_OUTPUT_PROTOCOL installed, and it cannot be a virtual device;
- Replace the Blt() interface function of EFI_GRAPHICS_OUTPUT_PROTOCOL with the BltRotate() function in GopRotate that implements the rotation display;
- The degree of rotation of the Shell interface is determined by the member variable Rotation of the internal private structure. Implement GRAPHICS_OUTPUT_PROTOCOL_ROTATE_PROTOCOL to control this variable. This Protocol provides two interface functions to get and set the Rotation value.

The private structure of the sample project GopRotate is shown in Example 1.

[Example 1] GopRotate 's private structure

```
typedef struct
{  UINTN Signature; //Private data structure signature  EFI_HANDLE Handle; //Handle for managing devices  EFI_GRAPHICS_OUTPUT_PROTOCOL_BLT Blt; //Original Blt() interface function  EFI_GRAPHICS_OUTPUT_PROTOCOL *Gop; //Gop instance  GRAPHICS_OUTPUT_PROTOCOL_ROTATE_PROTOCOL GopRotate;//Protocol used by users  ROTATE_SCREEN Rotation; //Control screen rotation angle } GRAPHICS_OUTPUT_ROTATE_PRIVATE; typedef enum {  Rotate0 = 0, //Do not rotate  Rotate90 = 1, //Rotate 90 degrees  Rotate180 = 2, //Rotate 180 degrees  Rotate270 = 3, //Rotate 270 degrees  RotateMax = 4 } ROTATE_SCREEN;
```

GRAPHICS_OUTPUT_PROTOCOL_ROTATE_PROTOCOL is used to set and get the screen rotation angle. It provides two interface functions, as shown in Example 2.

[Example 2] GRAPHICS_OUTPUT_PROTOCOL_ROTATE_PROTOCOL function interface

```
typedef struct _GRAPHICS_OUTPUT_PROTOCOL_ROTATE_PROTOCOL \
GRAPHICS_OUTPUT_PROTOCOL_ROTATE_PROTOCOL;
struct _GRAPHICS_OUTPUT_PROTOCOL_ROTATE_PROTOCOL
{  GRAPHICS_OUTPUT_PROTOCOL_ROTATE_GET_ROTATION GetRotation; //Get the rotation angle
GRAPHICS_OUTPUT_PROTOCOL_ROTATE_SET_ROTATION SetRotation; //Set the rotation angle }; typedef EFI_STATUS (EFIAPI
*GRAPHICS_OUTPUT_PROTOCOL_ROTATE_GET_ROTATION)(  IN GRAPHICS_OUTPUT_PROTOCOL_ROTATE_PROTOCOL *This, //Protocol instance
  IN ROTATE_SCREEN *Rotation //Angle of rotation  ); typedef EFI_STATUS (EFIAPI *GRAPHICS_OUTPUT_PROTOCOL_ROTATE_SET_ROTATION)(  IN
```

```
GRAPHICS_OUTPUT_PROTOCOL_ROTATE_PROTOCOL *This, //Protocol instance   IN ROTATE_SCREEN Rotation //Rotation angle   );
```

In order to realize the function of rotating the UEFI Shell interface, a large amount of code is implemented in the sample project GopRotate. These codes are mainly built around the private structure GRAPHICS_OUTPUT_ROTATE_PRIVATE to build logic. There are two core logics: one is to replace the original Blt() function, so that other UEFI applications or drivers actually call the function BltRotate() we prepared when calling the Blt() interface; the other is to realize the conversion display of the UEFI Shell interface according to the rotation angle specified by the user. This function is mainly implemented by the BltRotate() function.

The operation of replacing the interface function Blt() is implemented in the driver's Start() function, and the code will not be posted here.

After the operation of the Start() function, when the user calls the Blt() interface function, the BltRotate() function is actually called. In the BltRotate() function, the PerformTranslations() function is called, which processes the display according to different rotation angles.

In the GopRotate code, we can clearly see the processing logic. The function transforms the display according to the rotation angle set by the user. After completing the above two core logics, we can implement the two interface functions of setting the rotation angle and getting the rotation angle, as well as other management driver codes. These writing processes are very similar to the writing process of the framework driver BlankDrv, and are not difficult to understand. It is easy to understand by directly viewing the code.

2 UEFI application TestGopRotate

GopRotate provides GRAPHICS_OUTPUT_PROTOCOL_ROTATE_PROTOCOL and two interface functions of this protocol for users to set the rotation angle of the UEFI Shell interface and get the rotation angle. In order to test GopRotate, I wrote an application TestGopRotate, which can specify the rotation angle through the command line. The source code address is provided at the end of the article.

TestGopRotate can read the currently set rotation angle. Setting the rotation angle is achieved through different command line parameters. Its usage is shown in Example 3.

[Example 3] Usage of TestGopRotate

```
FS0:\> TestGopRotate    //Without parameters, get the current rotation angle
Rotate90 //Rotate 90 degrees
FS0:\> TestGopRotate 3 //You can choose parameters 0, 1, 2, 3, which represent rotation of 0, 90, 180 and 270 degrees respectively
```

The code writing method is similar to the writing method of the sample project TestServiceDrv in Chapter 67. Copy the header file GopRotate.h of the protocol to be accessed to the folder of TestGopRotate, and then write the code to access the protocol and interface functions. The core part of the code is shown in Example 4.

[Example 4] Set the rotation angle

```
EFI_STATUS Status;
GRAPHICS_OUTPUT_PROTOCOL_ROTATE_PROTOCOL *GopRotate = NULL;
  …… //Code omitted
Status=gBS->LocateProtocol(&gGraphicsOutputProtocolRotateProtocolGuid,
NULL, (VOID **)&GopRotate);//Get Protocol instance
  …… //Code omitted
if(Argc == 2)
{    switch(Argv[1][0])    {      case '0':          GopRotate->SetRotation(GopRotate, Rotate0); //Do not rotate        break;      case '1':        GopRotate-
>SetRotation(GopRotate, Rotate90); //Rotate 90 degrees        break;      case '2':        GopRotate->SetRotation(GopRotate, Rotate180); //Rotate 180 degrees
      break;      case '3':        GopRotate->SetRotation(GopRotate, Rotate270); //Rotate 270 degrees        break;      default:        break;    } } return
EFI_SUCCESS;
```

3 Testing

The method of compiling these two examples is the same as that in the previous articles.

Compile the driver:

*C:\UEFIWorkspace>build -t VS2015x86 -p RobinPkg\RobinPkg.dsc *
-m RobinPkg\Drivers\GopRotate\GopRotate.inf -a IA32

Compile UEFI application:

*C:\UEFIWorkspace>build -t VS2015x86 -p RobinPkg\RobinPkg.dsc *
-m RobinPkg\Applications\TestGopRotate\TestGopRotate.inf -a IA32

Operation effect:

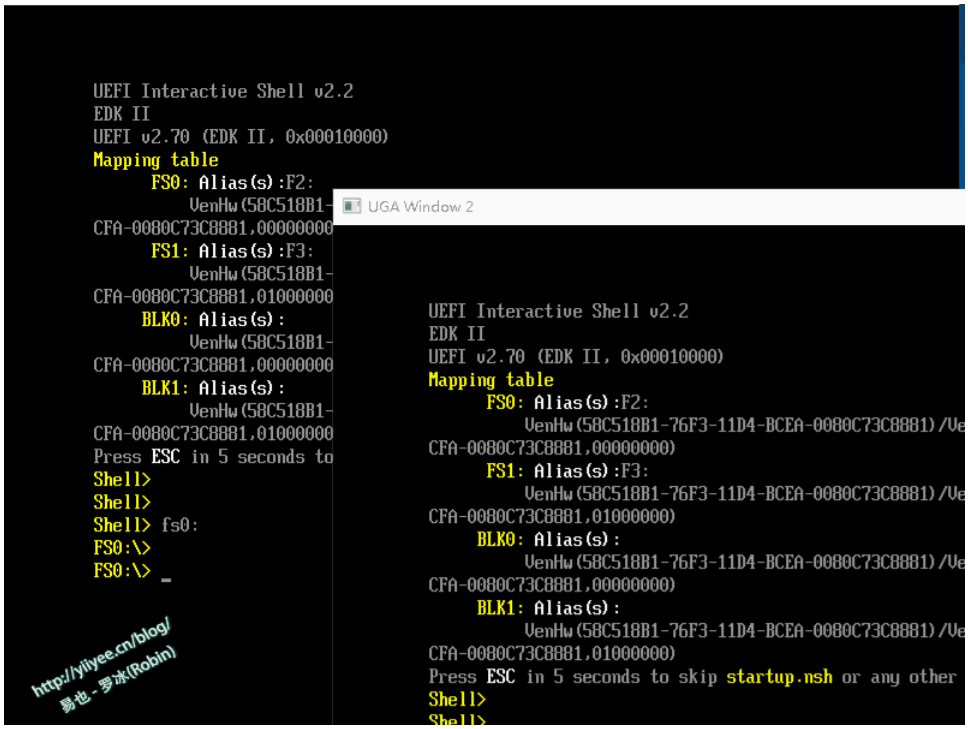

Figure 1 Testing GopRotate

Gitee address: https://gitee.com/luobing4365/uefi-explorer
Project code is located in: /FF RobinPkg/ RobinPkg /Applications/TestGopRotate
/FF RobinPkg/ RobinPkg /Drivers/GopRotate