

UEFI Development Exploration 67- YIE001PCIe Development Board (03 UEFI Driver)



luobing4365



Posted on 2021-01-14 19:33:29



Views: 826



collect



Likes 2

copyright

Category Column:

UEFI Development

Article Tags:

uefi

Low-level application development

bios

UEFI Drivers

UEFI Shell Commands



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 article, we completed a sample project for a service-type UEFI driver. In this article, we will write a UEFI application for testing to demonstrate how to use the protocol provided by the driver. In addition, we will also explain in detail how to load the driver for testing.

1Writing a test-driven UEFI application

After completing the construction of the service driver and the corresponding sample protocol, we are ready to write a UEFI application that accesses the sample protocol. The sample project TestServiceDrv prepared in this section provides a download address at the end of the article.

To write the code for the test example Protocol, follow these steps:

- Copy the header file MyProtocol.h of the sample project ServiceDrv to the folder of the sample project TestServiceDrv. At the same time, modify the TestServiceDrv.inf file and add this header file name under the [Sources] Section;
- Add the declaration and definition of GUID in the source file TestServiceDrv.c where the main program of TestServiceDrv is located;
- Add test code to the main program.

In the service driver example we prepared, three interface **functions** are prepared. Among them, the interface functions My_SampleIn() and MySample_DoSth() assign different values to private data, and the interface function My_SampleOut() prints out the private data. We can write test code based on this, as shown in Example 1.

[Example 1] Test Example Protocol

```
EFI_STATUS EFIAPI UefiMain (IN EFI_HANDLE ImageHandle,
IN EFI_SYSTEM_TABLE *SystemTable)
{ EFI_STATUS Status; EFI_MYSAMPLE_PROTOCOL *myprotocol; Status=gBS->LocateProtocol(&gEfiMYSampleProtocolGUID, //Example Protocol GUID
NULL, (VOID **)&myprotocol); if(EFI_ERROR(Status)) { Print(L"LocatProtocol error: %r\n", Status); return Status; } //Test the interface function of the
sample Protocol myprotocol->MySample_In(myprotocol, L"Hello, My protocol!"); myprotocol->MySample_Out(myprotocol); myprotocol-
>MySample_DoSth(myprotocol, L"Enjoy UEFI!"); myprotocol->MySample_Out(myprotocol); return EFI_SUCCESS; }
```

The sample application TestServiceDrv works with the service driver ServiceDrv built previously. If ServiceDrv is not loaded, it will prompt that the Protocol cannot be found; if ServiceDrv is loaded, it will call the interface functions of the sample Protocol in sequence as shown in Example 1.

2 Test-driven Shell Commands

The UEFI Shell commands used include load and dh. Their usage is described as follows.

- **The load** command is used to load the UEFI driver. Its syntax format is:

```
load [-nc] file [file...]
```

This command is used to load the UEFI driver into memory. It can process single or multiple driver files at a time. The file name supports wildcards. If the command is followed by the parameter "-nc", it means that the driver is only loaded into memory and not connected to the device. It is often used to load service drivers.

- **The dh** command is used to list the device handle information in the system, as well as device-related information, such as device path, driver name, etc. Its syntax format is:

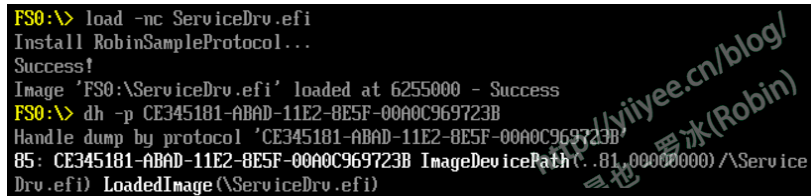
```
dh [-l <lang>] [handle | -p <port_id>] [-d] [-v]
```

Parameter description for this command:

-p <port_id>: List all handles with the specified GUID installed;
 -d: List information related to the UEFI driver;
 -l<lang>: Express in the specified language, such as ISO 639-2;
 -sfo: Use standard format output to display information;
 -v: Output verbose information;

handle: refers to the number of the UEFI handle in the system. If the handle is not specified when using this command, all handle information will be listed.

After compiling the sample project ServerDrv, you can use the load command in the simulator to load the driver and use the dh command to view the loaded status, as shown in Figure 1.



```

FS0:\> load -nc ServiceDrv.efi
Install RobinSampleProtocol...
Success!
Image 'FS0:\ServiceDrv.efi' loaded at 6255000 - Success
FS0:\> dh -p CE345181-ABAD-11E2-8E5F-00A0C969723B
Handle dump by protocol 'CE345181-ABAD-11E2-8E5F-00A0C969723B'
85: CE345181-ABAD-11E2-8E5F-00A0C969723B ImageDevicePath(C:\EFI\FS0\00000000)\Service
Drv.efi) LoadedImage(\ServiceDrv.efi)
  
```

Figure 1 Loading a service driver

3. Test Drive

When compiling ServiceDrv in the previous article, the target architecture was 32-bit. Therefore, the TestServiceDrv used for testing must also be a 32-bit architecture:

```

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

The test results are as follows:



```

Shell> fs0:
FS0:\> load -nc ServiceDrv.efi
Install RobinSampleProtocol...
Success!
Image 'FS0:\ServiceDrv.efi' loaded at 6355000 - Success
FS0:\> TestServiceDrv.efi
MySample_In.
User's string:Hello,My protocol!
mydata: myWord=0x11,myDowrd=0x1122
MySample_Out.
User's string:Enjoy UEFI!
Do something... MySample_DoStn.
mydata: myWord=0xAA,myDowrd=0x55
MySample_Out.
  
```

Figure 2 Using the Protocol provided by the TestServiceDrv test driver

Gitee address: <https://gitee.com/luobing4365/uefi-explorer>

Project code is located in: /FF RobinPkg/ RobinPkg /Applications/ TestServiceDrv

about Us Careers Business Cooperation Seeking coverage 400-660-0108 kefu@csdn.net Online Customer Service Working hours 8:30-22:00

Public Security Registration Number 11010502030143 Beijing ICP No. 19004658 Beijing Internet Publishing House [2020] No. 1039-165

Commercial website registration information Beijing Internet Illegal and Harmful Information Reporting Center Parental Control

Online 110 Alarm Service China Internet Reporting Center Chrome Store Download Account Management Specifications

Copyright and Disclaimer Copyright Complaints Publication License Business license

©1999-2025 Beijing Innovation Lezhi Network Technology Co., Ltd.