EFI Basic Tutorial (VIII) - Simple Use of PCD

原创 xiaopangzi313 Modified on 2024-10-07 00:57:28 Read 5.4k Collection 16 Likes 2

Category columns: 15_Firmware Development Article Tags: UEFI

摘要 This article introduces how to write and run a simple application in the UEFI environment, focusing on the use of the Platform Configuration Database (PCD), including the definition, reference, and method of obtaining the default value of the static PCD. An example is used to show how to define the PCD in MdeModulePkg.dec, reference it in MyHelloWorldPCD.inf, and use the PcdGetXX function in the C source cod…

The summary is generated in C Know , supported by DeepSeek-R1 full version, go to experience>

1. Write source code

1. Write the UEFI Application code C:\edkii\OvmfPkg\MyHelloWorldPCD\MyHelloWorldPCD.c,

```
EFI_STATUS
EFIAPI
MyHelloWorldPCDEntry(
  IN EFI_HANDLE        ImageHandle,
  IN EFI_SYSTEM_TABLE  *SystemTable
)
{
  EFI_STATUS  Status = EFI_SUCCESS;
  UINT32  PrintTimes ;
  UINT32  i ;
  CONST CHAR16 *PrintStr;
  // DEBUG ((EFI_D_ERROR , "[MyHelloWorldPCD] MyHelloWorldPCDEntry Start..\n"));
  Print(L"[MyHelloWorldPCD] MyHelloWorldPCDEntry Start..\n");

  if (FeaturePcdGet(PcdMyHelloWorldPrintEnable)){
    Print (L"[MyHelloWorldPCD] PcdMyHelloWorldPrintEnable ..\n");

    PrintTimes = PcdGet32(PcdMyHelloWorldPrintTimes);
    for (i = 0; i < PrintTimes; i++){
        PrintStr = PcdGetPtr(PcdMyHelloWorldPrintString);
        Print (L"[MyHelloWorldPCD]  Pcd  Str = %s\n",PrintStr);
    }
  }

  // DEBUG ((EFI_D_ERROR , "[MyHelloWorldPCD] MyHelloWorldPCDEntry End..\n"));
  Print(L"[MyHelloWorldPCD] MyHelloWorldPCDEntry End..\n");

  return Status;
}

```

2. Modify C:\code\local_edkrepo_10nm\Edk2\OvmfPkg\MyHelloWorldPCD\MyHelloWorldPCD.inf

```
#use to operate bool value
[FeaturePcd]
  gEfiMdeModulePkgTokenSpaceGuid.PcdMyHelloWorldPrintEnable  ## CONSUMES

[Pcd]
  gEfiMdeModulePkgTokenSpaceGuid.PcdMyHelloWorldPrintString  ## CONSUMES
  gEfiMdeModulePkgTokenSpaceGuid.PcdMyHelloWorldPrintTimes    ## SOMETIMES_CONSUMES
```

3. Modify C:\code\local_edkrepo_10nm\Edk2\MdeModulePkg\MdeModulePkg.dec

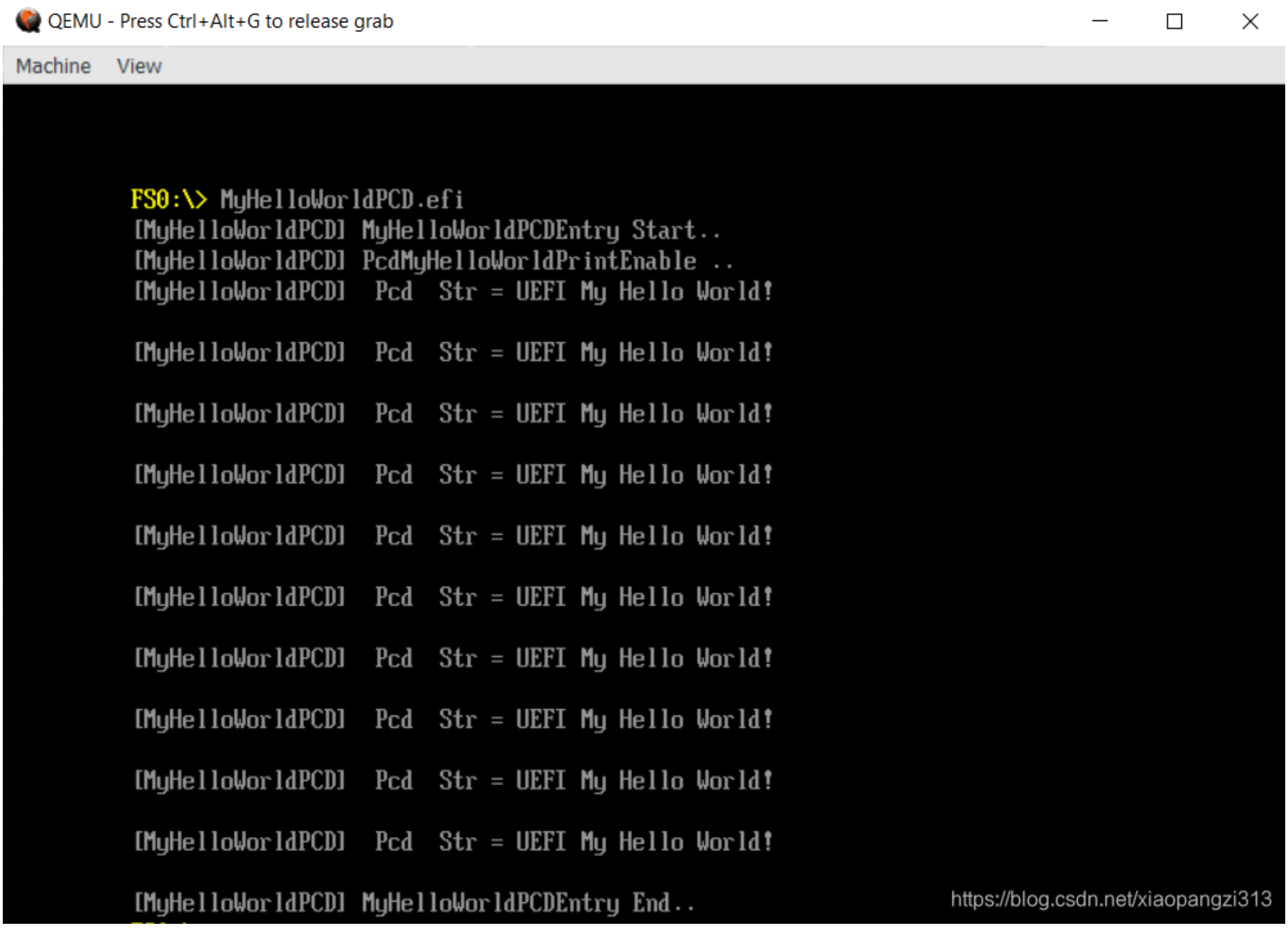
```
...
 [PcdsFeatureFlag]
  gEfiMdeModulePkgTokenSpaceGuid.PcdMyHelloWorldPrintEnable|TRUE|BOOLEAN|0x0001200d

 [PcdsFixedAtBuild, PcdsPatchableInModule, PcdsDynamic, PcdsDynamicEx]
  # @Prompt HellowWorld print times.
  gEfiMdeModulePkgTokenSpaceGuid.PcdHelloWorldPrintTimes|10|UINT32|0x40000005

  # @Prompt HelloWorld print string.
  gEfiMdeModulePkgTokenSpaceGuid.PcdHelloWorldPrintString|L"UEFI Hello World!\n"|VOID*|0x40000004
  ...
```

2. Compile and generate EFI files

Run and `edksetup.bat` compile the entire OvmfPkg Package

3. Run UEFI APP `MyHelloWorldPCD.efi`

1. Copy `C:\edkii\Build\OvmfX64\DEBUG_VS2013x86\FV\OVMF.fd` to `C:\qemu` ; Copy `C:\edkii\Build\OvmfX64\DEBUG_VS2013x86\X64\OvmfPkg\MyHelloWorldPCD\MyHelloWorldPCD\OUTPUT\MyHelloWorldPCD.efi` to virtual disk `HDD_BOOT.img`
2. Execute , and then execute in , the result is as follows, `setup-qemu-x64.bat  | findstr MyHelloWorldPCD UEFI SHELL MyHelloWorldPCD.efi`


QEMU - Press Ctrl+Alt+G to release grab
Machine
View
https://blog.csdn.net/xiaopangzi313

```
FS0:\> MyHelloWorldPCD.efi
[MyHelloWorldPCD] MyHelloWorldPCDEntry Start..
[MyHelloWorldPCD] PcdMyHelloWorldPrintEnable ..
[MyHelloWorldPCD]  Pcd  Str = UEFI My Hello World!

[MyHelloWorldPCD]  Pcd  Str = UEFI My Hello World!

[MyHelloWorldPCD]  Pcd  Str = UEFI My Hello World!

[MyHelloWorldPCD]  Pcd  Str = UEFI My Hello World!

[MyHelloWorldPCD]  Pcd  Str = UEFI My Hello World!

[MyHelloWorldPCD]  Pcd  Str = UEFI My Hello World!

[MyHelloWorldPCD]  Pcd  Str = UEFI My Hello World!

[MyHelloWorldPCD]  Pcd  Str = UEFI My Hello World!

[MyHelloWorldPCD]  Pcd  Str = UEFI My Hello World!

[MyHelloWorldPCD]  Pcd  Str = UEFI My Hello World!

[MyHelloWorldPCD] MyHelloWorldPCDEntry End..
```

IV. Summary

The PCD (Platform Config DataBase) block is used to define platform configuration data. Its purpose is to complete the configuration of the platform without changing the .inf file. PCD categories are divided into two categories:

1) PCDs that work during the compilation process are similar to global static variables in C language, but their values cannot be modified, including **PcdsFeatureFlag** , **PcdsFixedAtBuild** and **PatchableInModule** ; among them, FeatureFlag type PCD can only be defined as **Bool** value, FixedAtBuild can support multiple data types **UINIT32** , **UINT8** , **VOID** *, etc., **PcdsPatchableInModule** 编译 type PCD can be modified by GenPatchPcdTable in the stage, and 运行 its value can also be changed at the time (PatchableInModule essentially means that PCD is stored in the data segment of the EFI module).

2) Take effect during the runtime. This type of PCD includes **PcdsDynamicDefault** , **PcdsDynamicExDefault** , **PcdsDynamicHii** , **PcdsDynamicExHii** , **PcdsDynamicVpd** , **PcdsDynamicExVpd** , **PcdsDynamic , and PcdsDynamicEx .**
Among them, PcdsDynamicDefault and PcdsDynamicExDefault can be changed during the Runtime phase, but the change value will be lost when the memory is powered off. The format is as follows:

```
[PcdsDynamic]
gEfiCpuTokenSpaceGuid.PcdCpuProcessorFeatureCapability|0|UNIT32|0x4000002
```

The PCD value is 0, the type is UNIT32, and the token number is 0x4000002.

PcdsDynamicHii and **PcdsDynamicExHii** are used to define the default variable. When an EFI variable is undefined, a new non-volatile variable will be created and the default value will be set. The format is as follows:

```
[ PcdsDynamicHii.common.DEFAULT]
gEfiIntelFrameworkModulePkgTokenSpaceGuid.PcdPlatformBootTimeOut|L"Timeout"|gEEfiGlobalVariableGuid|0x0 | 5
```

In this example, the Hii Variable name is set to "TimeOut", the Variable Guid is set to gEEfiGlobalVariableGuid, the Variable Offset is set to 0, and the default value is set to 5 (PcdsDynamicHii will eventually call the Variable service).

The PCD variables modified by PcdsDynamicVpd and **PcdsDynamicExVpd** will be stored in the VPD database (located on the Flash), and the operation of PCD will be directly mapped to the VPD area on the Flash. The format is as follows:

AI generated projects 登录复制

```
[PcdsDynamicVpd.common.DEFAULT]
gEfiE6xxTokenSpaceGuid.PcdIgdPreAllocSize | 4 | 0x2
```

In the example, Pcd VpdOffset is 4, and the initial value is 2 (VPD type PCD cannot use PCD Set interface).
PcdsDynamic is similar to PcdsDynamicEx and PcdsDynamicDefault and PcdsDynamicExDefault. If a PCD module is integrated into PlatformBuild, **PcdsDynamic is equivalent to PcdsDynamicEx** and PcdsDynamicDefault, that is, it is called in the source code `PcdGetxx(PcdSampleDynamicPcd)` ; if the module is released as an independent binary, **PcdsDynamicEx** is used , that is, it is called in the source code `PcdGetxx(gEfiMyTokenSpaceGuid, PcdSampleDynamicPcd)` .

The program in this article mainly implements the use of static PCD in a simple way. It defines PcdMyHelloWorldPrintEnable, PcdHelloWorldPrintTimes, and PcdHelloWorldPrintString in the MdeModulePkg.dec file, references PCD in the MyHelloWorldPCD.inf file, and finally uses the PcdGetXX function in MyHelloWorldPCD.c to obtain the default value of PCD and perform simple processing.

PCD DEMO source code