UEFI Development Exploration 77- YIE001PCIe Development Board (10 toggle switches and display)

原创 luobing4365 Posted on 2021-02-13 12:09:56 Read 470 collect Likes

Category Column: UEFI Development Article Tags: UEFI Low-level application development PCI-E UEFI Drivers Option ROM

YIE001PCIe development board toggle switch

There are two toggle switches on the YIE001. Their status can be obtained through the corresponding register bits to achieve interactive control. This article attempts to obtain the key and the coding of the graphic display.

1 Get the state of the toggle switch

In the 75th chapter of UEFI Development Exploration, the hardware resources of the YIE001 development board are listed. The board provides two toggle switches, corresponding to GPI1 and GPI2 of CH366.

Refer to the CH366 chip manual to define the relevant macro definitions:

```
#define CH366GPIR 0x02

#define KEY1 1   //GPI1 -- GPIR(IO基地址+02h) bit 1
#define KEY2 2   //GPI2 -- GPIR(IO基地址+02h) bit 2
```

On the development board YIE001, when the toggle switch is turned down (that is, towards the PCIE slot), the corresponding register value is 0; when it is turned up, the corresponding register value is 1. The implementation code is as follows:

```
/**
  获取开发板YIE001上拨动开关的状态.

  @param  IoBaseAddress     YIE001上PCIE芯片的基地址
  @param  KeyNum            拨动开关的标识，Key1和Key2
  @retval 1           拨下（靠近PCIE插槽）
          0           拨上
**/
UINT8 GetYIE001Key(UINT16 IoBaseAddress,UINT8 KeyNum)
{
  UINT8 regValue=0;

  regValue = IoRead8(MyIoBaseAddr+CH366GPIR);   //GPIR
  if(KeyNum == KEY1)
    regValue &=0x02;
  if(KeyNum == KEY2)
    regValue &=0x04;
  if(regValue)
    return 1;
  else
    return 0;
twen
twen }
```

2 Display switch status

In order to display the status of the toggle switch, the code in the previous blog is reused. The example project of this blog is YIE1Key, which combines the code for accessing the YIE001 development board and the graphic display.

It is easier to understand the functions implemented by looking at the code:

```
VOID HelloMyROM(VOID)
{

```

```
  EFI_INPUT_KEY key={0,0};
  UINT64 flag;
  UINT8 *s_text1 = "欢迎进入UEFI的世界！";
  UINT8 *s_text2 = "按'ESC'键或'1'退出此界面";
  UINT8 *s_key1str1 = "拨动开关1拨下！";
  UINT8 *s_key1str2 = "拨动开关1拨上！";
  UINT8 *s_key2str1 = "拨动开关2拨下！";
  UINT8 *s_key2str2 = "拨动开关2拨上！";
  UINT8 tempKey1;
  UINT8 tempKey2;
  //图形显示测试
  flag = InintGloabalProtocols(GRAPHICS_OUTPUT);
    Print(L"flag=%x\n",flag);
  SwitchGraphicsMode(TRUE);
    SetBKG(&(gColorTable[DEEPBLUE]));
  draw_string(s_text1, 120, 100, &MyFontArray, &(gColorTable[YELLOW]));
  draw_string(s_text2, 120, 140, &MyFontArray, &(gColorTable[YELLOW]));
  tempKey1 = GetYIE001Key(MyIoBaseAddr,KEY1);
  tempKey2 = GetYIE001Key(MyIoBaseAddr,KEY2);
  if(tempKey1)
    draw_string(s_key1str1, 120, 200, &MyFontArray, &(gColorTable[WHITE]));
  else
    draw_string(s_key1str2, 120, 200, &MyFontArray, &(gColorTable[WHITE]));
  if(tempKey2)
    draw_string(s_key2str1, 120, 230, &MyFontArray, &(gColorTable[WHITE]));
  else
    draw_string(s_key2str2, 120, 230, &MyFontArray, &(gColorTable[WHITE]));
  while(key.ScanCode!=0x17) //ESC
  {
    GetKey(&key);
    if(key.UnicodeChar == 0x31)
      break;
    if(GetYIE001Key(MyIoBaseAddr,KEY1)!=tempKey1)
    {
      if(tempKey1)
      {
        tempKey1 = 0;
        draw_string(s_key1str1, 120, 200, &MyFontArray, &(gColorTable[DEEPBLUE]));  //消除显示
        draw_string(s_key1str2, 120, 200, &MyFontArray, &(gColorTable[WHITE]));
        SetLed(MyIoBaseAddr,LED1,LEDON);
      }
      else
      {
        tempKey1 = 1;
        draw_string(s_key1str2, 120, 200, &MyFontArray, &(gColorTable[DEEPBLUE]));  //消除显示
        draw_string(s_key1str1, 120, 200, &MyFontArray, &(gColorTable[WHITE]));
        SetLed(MyIoBaseAddr,LED1,LEDOFF);
      }
      Delayms(200);
    }
    if(GetYIE001Key(MyIoBaseAddr,KEY2)!=tempKey2)
    {
      if(tempKey2)
      {
        tempKey2 = 0;
        draw_string(s_key2str1, 120, 230, &MyFontArray, &(gColorTable[DEEPBLUE]));  //消除显示
        draw_string(s_key2str2, 120, 230, &MyFontArray, &(gColorTable[WHITE]));
        SetLed(MyIoBaseAddr,LED2,LEDON);
      }
      else
      {
        tempKey2 = 1;
        draw_string(s_key2str2, 120, 230, &MyFontArray, &(gColorTable[DEEPBLUE]));  //消除显示
        draw_string(s_key2str1, 120, 230, &MyFontArray, &(gColorTable[WHITE]));
        SetLed(MyIoBaseAddr,LED2,LEDOFF);
      }
      Delayms(200);
    }
  }
  SetMyMode(OldGraphicsMode);
  SwitchGraphicsMode(FALSE);
}
```

The HelloMyROM() function in the framework code has been modified . Its main functions are as follows:

(1) Initialize the graphic display and print the corresponding prompt string on the screen;

(2) Enter the key acquisition loop and exit the loop only when the user presses 'ESC' or '1';

(3) Get the toggle switch status and update the display of the toggle switch status.

3 Testing

The compilation command is as follows:

```
C:\UEFIWorkspace>build -t VS2015x86 -p RobinPkg\RobinPkg.dsc -m RobinPkg\Drivers\YIE1Key\YIE1Key.inf -a X6
```

The interface of Option ROM is shown in Figure 1.

Figure 1 The interface of YIE1Key

After flipping the corresponding switch, the interface will display the current state of the switch. Combining this flip switch interaction mechanism of YIE001, you can try to achieve richer and more interesting effects.

Gitee address: https://gitee.com/luobing4365/uefi-explorer
The ROM file used in the project is located at: /77 YIE1Key