Applieddata.net

Embedded Computer Systems

Adding Device Drivers to an ADS System

ADS Document 110010-10111

Introduction

Loading a device driver that is not included in ROM is a fairly simple task. It involves ensuring that the driver and its associated dependencies are available when the system starts, and that the registry entries required for the driver are available for driver initialization.

This article will walk through the process of installing a driver by setting up a BitsyX system to load a simple LED driver. The sample driver, will full source code, is available as ADS part number 720020-11752. While this is an example specific to a driver and piece of hardware, the process for other systems and hardware will be the same.

Step 1 – Verify the System Path

Windows CE by default will look for drivers in the \Windows folder. The system can also be instructed to look in other locations by setting the SystemPath in the device registry by adding values to the following key:

```
[HKEY_LOCAL_MACHINE\Loader]
      "SystemPath"=multi_sz:”{path(s)}”
```

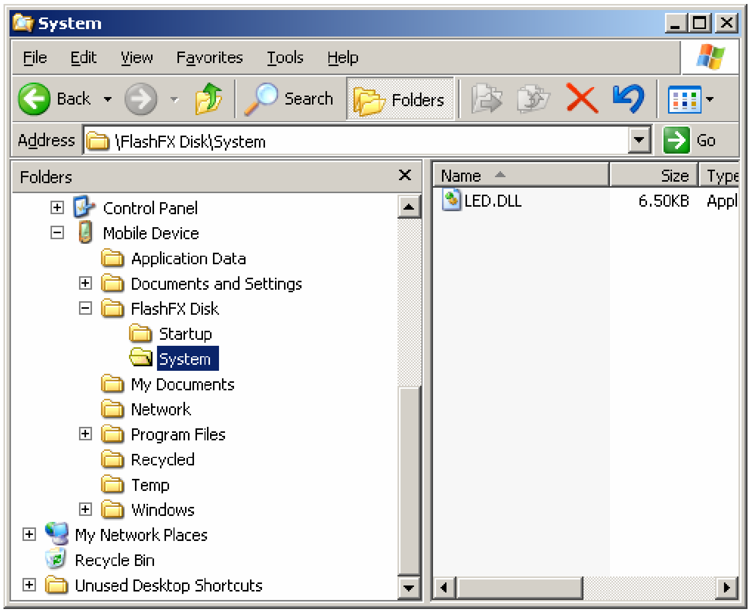
All newer ADS systems have already extended the SystemPath to include the following device folders.

```
\FlashFX Disk\System
\Storage Card\System
```

Note that the registry key is a multi-sz (multiple string) type, which the neither on-board registry editor or ADSLOAD.REG can alter. The built-in editor can be used to view the first string value in the key. The Remote Registry Editor tool in eMbedded Visual C++ can be used to view all values in this key.

Step 2 – Store the Driver on the System

Once you're verified that the system path supports a persistent folder, simply copy the driver and any necessary dependencies to that folder. In our example our driver binary is “led.dll” and we'll store it in the on-board flash under \FlashFX Disk\System as you can see in the following screen capture:



Address \FlashFX Disk\System

Name	Size	Type
LED.DLL	6.50KB	Appl

Step 3 – Set the Driver Registry Entries

Next we need to provide the registry entries the driver will need, and they need to be persistent. The simplest way to do this is to append the entries to your device's ADSLOAD.REG file and re-burn it to flash. For more detail on updating ADSLOAD.REG see the "ADS Windows CE Getting Started Guide."

Most drivers will require four key values at a minimum, and our LED driver is no exception. The values we need are:

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\LedDriver]
      "Dll"="led.dll"
      "Prefix"="LED"
      "Index"=dword:1
      "Order"=dword:9
```

These are the settings used by device.exe to load our driver at boot.

Step 4 – Test the Installation

The only remaining step is to test your installation. Our sample LED driver is easy to test as it outputs a debug message when it is loaded by device.exe, so we don't even need an application.

Here is a capture some of the debug output from the device before we installed our driver:

```
...
SMARTIO Detected : PartType=0x8535 Firmware=0x0010
Revision=2
SMARTIO : SerialClockRate=10 KBits/S(0xa7).
Initializing Legacy I2C Driver...ok
Initializing I2C driver v 1.0...ok
Initializing Digital I/O Driver...ok

   Microsoft Windows CE Ver. 4.20 (build 0)

   BitsyX Windows CE Ver. 4.20.17 BITSYX_CORE_RAM
```

And here is a capture of the same output after installation:

```
...
SMARTIO Detected : PartType=0x8535 Firmware=0x0010
Revision=2
SMARTIO : SerialClockRate=10 KBits/S(0xa7).
Initializing Legacy I2C Driver...ok
Initializing I2C driver v 1.0...ok
Initializing Digital I/O Driver...ok
Initializing LED Driver...ok

   Microsoft Windows CE Ver. 4.20 (build 0)

   BitsyX Windows CE Ver. 4.20.17 BITSYX_CORE_RAM
```

Notice the bolded line, which is the output from the LED_Init function of our driver. This shows that our driver is loading on every device boot.